



Trabajo Fin de Máster

Control multi-robot para el transporte de objetos deformables

Multi-robot control for transport of deformable objects

Autor

Raquel Marcos Saavedra

Directores

Gonzalo López Nicolás

Miguel Aranda Calleja

Titulación

Máster en Ingeniería Industrial

AGRADECIMIENTOS

En primer lugar, gracias a Miguel y Gonzalo por su absoluta dedicación, su ayuda en cada dificultad que se presentaba y la confianza depositada en mí.

Gracias a mis padres, Carlos y Conchi, por su amor incondicional, sus consejos y su constante apoyo, por estar siempre a mi lado, acompañándome en el camino, y por levantarme siempre que me caigo. Porque sin ellos no sería lo que soy hoy.

Gracias a Nico por creer siempre en mí y celebrar mis logros como si fuesen propios, por acompañarme en los momentos difíciles y tener siempre la palabra justa para animarme. Por las risas, por ser mi pilar y por hacer que la vida sea mucho más bonita.

Gracias a mi familia y amigos por todos los momentos vividos.

Por último, agradecer al Instituto de Investigación en Ingeniería de Aragón (I3A) y a la Cátedra de Transformación Industrial (Universidad de Zaragoza - Gobierno de Aragón) por haber financiado esta investigación.



RESUMEN

Control multi-robot para el transporte de objetos deformables

El transporte de objetos deformables mediante equipos de múltiples robots móviles es una tarea compleja que requiere un control muy preciso y coordinado. Por ello, el presente Trabajo de Fin de Máster se centra en el desarrollo, implementación y análisis de un controlador avanzado para sistemas multi-robot, modelados mediante dinámicas de integrador simple, cuyo fin es transportar objetos deformables hacia una configuración deseada. Durante la tarea se controla la forma, escala, posición y orientación de la formación, restringiendo las deformaciones a modos lineales y cuadráticos. Estas restricciones permiten conservar la integridad del objeto, además de hacer el sistema más flexible.

El trabajo se estructura en dos partes fundamentales. La primera se basa en el desarrollo del algoritmo de control y su implementación en entornos 2D. La segunda parte extiende este control a espacios 3D, donde los robots tienen mayor libertad de movimiento. En este contexto, se implementan y comparan dos técnicas de control de rotación: la aproximación mediante rotaciones infinitesimales y el algoritmo de Kabsch. Para la evaluación experimental del controlador desarrollado se han diseñado diversos escenarios, ejecutados en simulación con MATLAB y con robots físicos en la plataforma robótica remota Robotarium, ubicada en el Instituto de Tecnología de Georgia. El resultado final de esta investigación es la validación exitosa del algoritmo de control, demostrando su efectividad tanto en entornos 2D como 3D.

Este trabajo se ha realizado dentro del Programa de Becas y Ayudas del Instituto de Investigación en Ingeniería de Aragón (I3A), en colaboración con la Cátedra de Transformación Industrial (Universidad de Zaragoza - Gobierno de Aragón).

Índice

1. Introducción	1
1.1. Objetivos	2
1.2. Contribuciones	3
1.3. Planificación	4
1.4. Estructura de la memoria	4
1.5. Nomenclatura	5
2. Estado de la materia	7
2.1. Control de forma de equipos multi-robot	7
2.2. Modelos basados en modos de deformación	8
2.3. Robotarium	9
3. Control multi-robot con modos de deformación en entornos 2D	11
3.1. Control de la deformación	12
3.1.1. Control de forma	12
3.1.2. Control de escala	16
3.2. Control de traslación y control de rotación	17
3.3. Métricas de error	18
4. Evaluación experimental en entornos 2D	19
4.1. Implementación en Robotarium	19
4.2. Transporte de objeto deformable en entorno 2D sin obstáculos	21
4.3. Transporte de objeto deformable en entorno 2D con obstáculo	23

4.4. Transporte de objeto deformable en entorno 2D por pasillo curvo	25
5. Extensión del controlador a entornos 3D	29
5.1. Control con rotaciones infinitesimales	30
5.1.1. Control de forma	30
5.1.2. Control de escala, posición y orientación	32
5.2. Control con algoritmo de Kabsch	32
5.2.1. Control de forma	33
5.2.2. Control de escala, posición y orientación	34
5.3. Métricas de error en entornos 3D	34
6. Evaluación experimental en entornos 3D	37
6.1. Transporte de objeto deformable en entorno 3D sin rotación	38
6.2. Transporte de objeto deformable en entorno 3D con rotación	40
6.3. Transporte de objeto deformable volumétrico	42
7. Conclusiones y trabajo futuro	45
7.1. Conclusiones alcanzadas	45
7.2. Trabajo futuro	46
Bibliografía	49
Anexos	52
A. Workshop 2024	55
B. Ejemplos de deformaciones alcanzables	61
C. Resultados gráficos en entornos 2D	65
C.1. Gráficas de la simulación 4.2	65
C.2. Gráficas de la simulación 4.3	69
C.2.1. Conservación de forma durante el transporte	75

C.3. Gráficas del experimento 4.4	76
C.3.1. Resultados obtenidos en simulación	76
C.3.2. Resultados obtenidos con robots reales	83
C.3.3. Justificación de la necesidad de <i>waypoints</i>	85
C.4. Transporte con diferente número de robots	86
C.4.1. Demostración de control de escala desacoplado	87
D. Resultados gráficos en entornos 3D	89
D.1. Gráficas de la simulación 6.1	89
D.1.1. Resultados cuando no se controlan las deformaciones	95
D.2. Gráficas de la simulación 6.2	98
D.2.1. Resultados cuando no se controla la rotación	100
D.3. Gráficas de la simulación 6.3	102
E. Fragmentos de código	105
F. Lista de videos	111
F.1. Vídeo de transporte en entorno 2D sin obstáculos	111
F.2. Vídeo de transporte en entorno 2D con obstáculo	112
F.2.1. Vídeo de conservación de forma durante el transporte	112
F.3. Vídeo de transporte en entorno 2D por pasillo curvo	113
F.3.1. Vídeo de simulación	113
F.3.2. Vídeo con robots reales	113
F.4. Vídeo de transporte en entorno 2D con diferente número de robots . . .	114
F.5. Vídeo de transporte en entorno 3D sin rotación	114
F.6. Vídeo de transporte en entorno 3D con rotación	115
F.7. Vídeo de transporte de objeto deformable volumétrico	115

Capítulo 1

Introducción

El crecimiento de la robótica durante las últimas décadas ha transformado profundamente múltiples sectores de la sociedad y la industria. Desde las «tortugas» de Grey Walter, capaces de ejecutar acciones simples de forma autónoma, hasta los complejos sistemas de múltiples robots actuales, la robótica móvil ha ido evolucionando hasta convertirse en una disciplina clave dentro de la ingeniería y la tecnología [1].

Uno de los grandes sectores que más interés ha suscitado recientemente es el control de sistemas de múltiples robots móviles para realizar tareas complejas de forma precisa y coordinada [2]. En particular, la manipulación y el transporte de objetos deformables por parte de equipos multi-robot es un área de investigación muy extendida que se enfrenta a continuos desafíos.

El presente Trabajo de Fin de Máster, desarrollado en colaboración con el Instituto de Investigación en Ingeniería de Aragón (I3A) y la Cátedra de Transformación Industrial (Universidad de Zaragoza - Gobierno de Aragón), pretende continuar la línea de trabajo del Trabajo de Fin de Grado «Control de forma multi-robot en el Robotarium» [3], aportando soluciones avanzadas para el transporte multi-robot de objetos deformables, tanto superficiales como volumétricos (Figura 1.1). Este tipo de tareas requieren un control estricto, de manera que se logre transportar el objeto sin que este sufra daños.



(a) Objeto superficial



(b) Objeto volumétrico

Figura 1.1: Ejemplos de objetos deformables que pueden ser transportados por equipos multi-robot.

La problemática tratada a lo largo de las siguientes páginas se centra en el control coordinado de un equipo de robots móviles encargados de transportar un objeto deformable desde una configuración inicial a otra final, asegurando que, siempre que sea posible, el objeto mantenga su forma. En aquellas situaciones donde, por las condiciones del entorno, la formación robótica deba cambiar de forma, lo hará de manera controlada según modos de deformación lineales y cuadráticos. Esta idea se refleja gráficamente en la Figura 1.2 para una mejor comprensión.

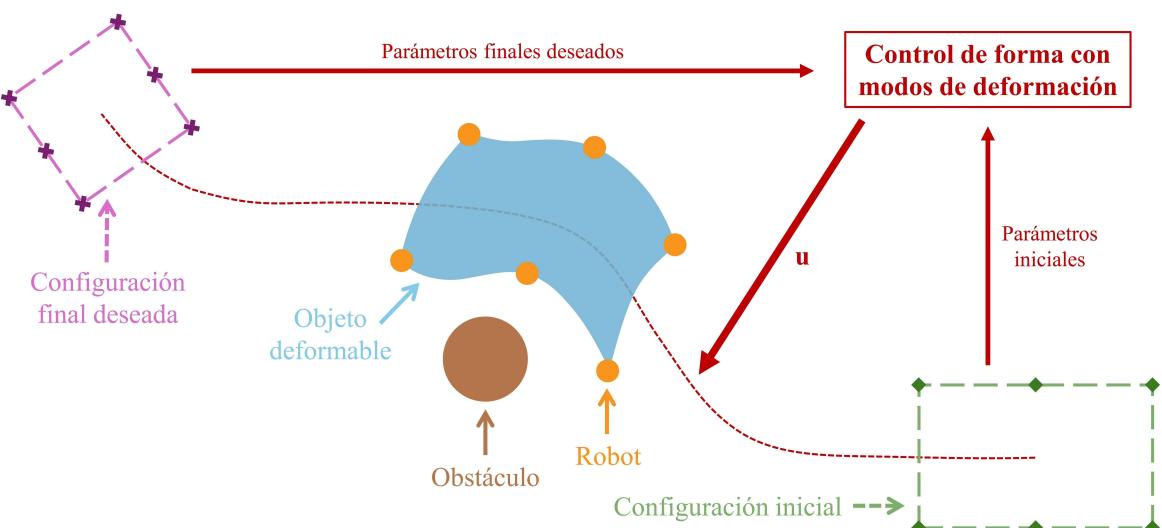


Figura 1.2: Esquema orientativo de la aplicación del control de forma con modos de deformación (u) a un equipo de robots que transporta una lámina flexible.

1.1. Objetivos

El objetivo principal de este TFM es implementar y validar el controlador antes mencionado mediante el diseño de un algoritmo que modele una formación de múltiples robots móviles con dinámicas de integrador simple y controle los cuatro parámetros que definen la configuración del equipo (forma, escala, posición y orientación).

El trabajo se divide en dos partes fundamentales. La primera de ellas se centra en el control de la formación robótica en un entorno bidimensional. Se desarrollarán diferentes estrategias y escenarios que permitirán verificar el correcto funcionamiento del controlador en simulaciones de MATLAB y en la plataforma Robotarium [4]. En la segunda parte, se extiende este control al entorno tridimensional, donde la complejidad aumenta significativamente debido a una mayor libertad de traslación y rotación por parte del equipo de robots. Para abordar este desafío, se implementarán y compararán dos métodos de control de rotación: el algoritmo de Kabsch y la aproximación mediante rotaciones infinitesimales. Los experimentos aquí realizados se validarán por medio de simulaciones en MATLAB.

Como objetivo personal, se busca ampliar conocimientos en el área del control de robots para la manipulación y transporte de objetos deformables, así como profundizar en el ámbito de la investigación científica.

1.2. Contribuciones

Como resultado de la primera parte del TFM, basada en el control multi-robot en entornos 2D, se ha publicado un artículo de investigación en el *4th Workshop on Representing and Manipulating Deformable Objects* del congreso internacional *IEEE International Conference on Robotics and Automation* (ICRA) de 2024 [5] [6]. En trabajos anteriores se empleaban modos de deformación lineales. Los modos cuadráticos propuestos en el presente trabajo proporcionan una mayor flexibilidad y un control más refinado de la deformación durante la tarea multi-robot. El artículo mencionado se muestra en el Anexo A.

La segunda parte del TFM extiende el uso de modos de deformación a entornos 3D. Además, se propone una formulación novedosa del control multi-robot basada en rotaciones infinitesimales que, comparada con métodos ya existentes, resulta más simple de tratar y evita el uso de algoritmos numéricos durante la ejecución.

Todo el trabajo plasmado en esta memoria contribuye a sentar las bases para futuras investigaciones sobre la manipulación y el transporte de objetos deformables mediante equipos de robots, ámbito de gran relevancia en diversas industrias como la manufacturera, la logística o, incluso, la biomédica.

1.3. Planificación

El desarrollo de este proyecto se ha realizado en varias fases bien definidas que han permitido abordar con éxito los objetivos planteados.

- Programación de un algoritmo de control que asegure la conservación de la forma del objeto durante su transporte en el espacio 2D, permitiendo deformaciones cuadráticas en aquellas situaciones donde las condiciones del entorno lo requieran.
- Diseño de experimentos para el análisis y validación del correcto funcionamiento del controlador. Simulación en MATLAB y ejecución real en Robotarium.
- Adaptación del controlador a entornos 3D, abordando la complejidad adicional del movimiento en tres dimensiones.
- Implementación de dos métodos de control de rotación tridimensional: el algoritmo de Kabsch y las rotaciones infinitesimales. Comparación de su efectividad y aplicabilidad.
- Simulación en MATLAB de diferentes experimentos de validación de los métodos desarrollados en entornos 3D y análisis de los parámetros más relevantes.
- Recopilación de todo el trabajo realizado y redacción de la memoria, exponiendo las conclusiones alcanzadas.

1.4. Estructura de la memoria

A continuación, se presenta una breve descripción de la estructura de la memoria, destacando los contenidos y objetivos de cada uno de los capítulos que la componen.

- El *Capítulo 1* plantea el contexto, alcance y objetivos del proyecto.
- En el *Capítulo 2* se exponen, de forma resumida, herramientas e investigaciones previas en las que se fundamenta el trabajo realizado.
- En el *Capítulo 3* se define el marco teórico en el que se basa el algoritmo de control en entornos 2D, y en el *Capítulo 4* se presenta la información relativa a los experimentos realizados con dicho controlador y el análisis de los mismos.

- El *Capítulo 5* muestra la adaptación del controlador a entornos 3D, así como la implementación de las dos técnicas de control de rotación. Los resultados de las simulaciones realizadas se recogen en el *Capítulo 6*.
- Por último, el *Capítulo 7* reúne todas las conclusiones alcanzadas y posibles futuras líneas de investigación.
- Esta memoria se completa con una serie de Anexos que ofrecen información más detallada sobre algunos aspectos del proyecto, además de algunos resultados adicionales.

1.5. Nomenclatura

Este apartado pretende aunar la nomenclatura básica utilizada a lo largo de la memoria para facilitar la interpretación y consulta de la misma. Como comentario adicional, notar que las magnitudes escalares están escritas en letra normal, los vectores en minúscula y negrita, y las matrices en mayúscula y negrita.

- \otimes representa el producto de Kronecker.
- \dot{p} denota la derivada temporal de p .
- $\|\cdot\|$ es la norma euclídea y $\|\cdot\|_F$ la norma de Frobenius.
- $(\cdot)^\top$ indica la traspuesta de una matriz.
- $(\cdot)^+$ representa la pseudoinversa de Moore-Penrose.
- $\text{tr}(\cdot)$ es la traza de una matriz.
- $\text{atan2}(y, x)$ es la función arcotangente de los parámetros x e y .
- $\text{vec}(\mathbf{A}) \in \mathbb{R}^{mn \times 1}$ es un vector columna que se obtiene al apilar las columnas de la matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$.
- $\mathbf{1}_N$ es un vector columna de N unos.
- \mathbf{I}_N es la matriz identidad de dimensiones $N \times N$.
- $\mathbf{S} = [[0, 1]^\top, [-1, 0]^\top]$ es una matriz de rotación de 90° en sentido antihorario.
- $[\mathbf{a}]_\times = [[0, a_3, -a_2]^\top, [-a_3, 0, a_1]^\top, [a_2, -a_1, 0]^\top]$ denota la matriz antisimétrica del vector $\mathbf{a} = [a_1, a_2, a_3]^\top$.

Capítulo 2

Estado de la materia

Este capítulo introduce algunas investigaciones y trabajos previos relevantes en la manipulación y transporte de objetos deformables, sobre los cuales se construye este Trabajo de Fin de Máster. También se presenta la plataforma Robotarium, cuyo uso ha sido imprescindible para alcanzar algunos de los objetivos planteados.

2.1. Control de forma de equipos multi-robot

Transportar un objeto deformable puede resultar complicado, sobre todo si es grande, pesado o frágil. Parece lógico pensar que, si la tarea se realiza mediante un equipo de robots, el transporte será más sencillo. En cambio, esto conlleva desafíos adicionales, como la necesidad de asegurar una elevada coordinación entre agentes. El control de forma permite que estos sistemas multi-robot manipulen dichos objetos con movimientos muy precisos, garantizando, en todo momento, la integridad del objeto.

La manipulación de objetos deformables con múltiples robots es un campo ampliamente tratado en trabajos anteriores [7] e implementado en diferentes escenarios [8]-[12]. Un ejemplo de ello es el trabajo realizado por Aranda *et al.* [13], que combina la traslación del equipo con la conservación de forma y las transformaciones afines respecto a una configuración de referencia. Esto permite que la formación robótica, modelada mediante dinámicas de integrador simple, pueda ejecutar un movimiento coordinado y eficiente.

Otros estudios ofrecen un enfoque alternativo asumiendo dinámicas de doble integrador [14] para introducir el efecto de la inercia en el sistema. El objetivo del controlador propuesto por Herguedas *et al.* es dirigir un equipo de múltiples robots que transportan

un objeto deformable desde una configuración inicial a otra final, definida por la combinación de una forma, escala, posición y orientación deseadas, en el entorno 2D. Este artículo fue el pilar fundamental del Trabajo de Fin de Grado «Control de forma multi-robot en el Robotarium» [3], en el cual se implementó el controlador mencionado y se verificó su correcto funcionamiento en diversos entornos. En la Figura 2.1 se muestra un ejemplo sencillo de la aplicación de este control.

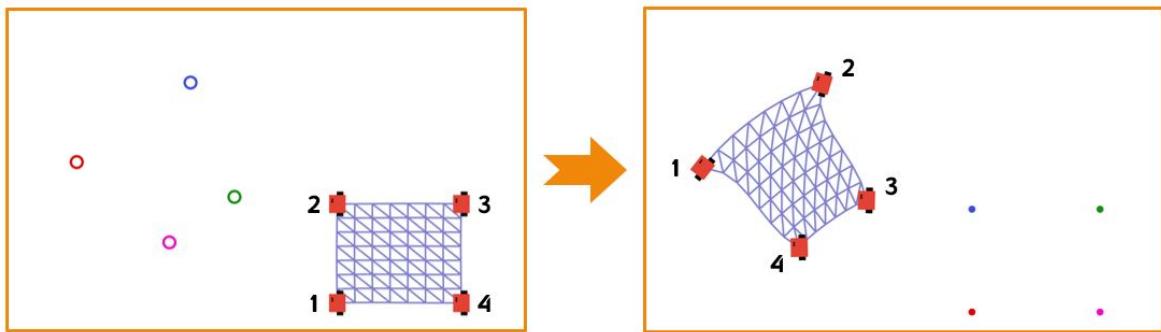


Figura 2.1: Ejemplo sencillo donde un equipo de robots transporta una lámina flexible desde una configuración inicial hasta una configuración final deformada, resultado de implementar el controlador propuesto en [14].

2.2. Modelos basados en modos de deformación

El análisis modal a menudo se utiliza para el modelado de estructuras en distintos contextos. Un ejemplo de ello es su gran relevancia en la computación gráfica para modelar objetos deformables. Dos ejemplos de investigaciones en este campo son el trabajo de Pentland y Williams [15], donde se utiliza el Método de Elementos Finitos, y el trabajo de Müller *et al.* [16], que plantea un método basado en el *shape matching* y estudia las deformaciones lineales, cuadráticas y mixtas que puede sufrir un objeto.

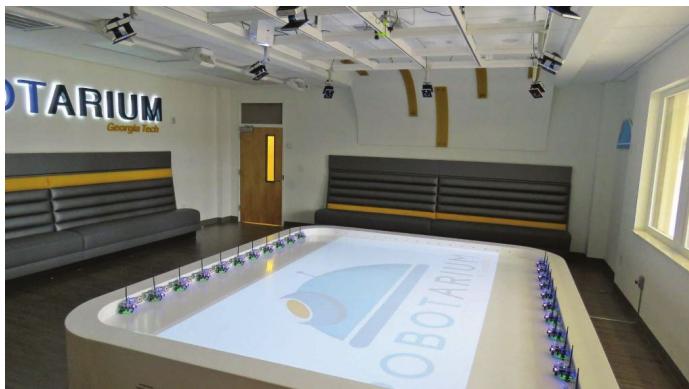
Este último enfoque puede extenderse a aplicaciones en robótica, permitiendo estimar las diversas maneras en las que un objeto puede deformarse al ser manipulado por los robots, tal como exponen Güler *et al.* [17]. Otros trabajos que abordan conceptos relacionados con la temática tratada a lo largo de esta investigación son [18]-[22].

En los siguientes capítulos se detalla la relación específica de estos estudios con el presente trabajo.

2.3. Robotarium

Robotarium nace a manos del Instituto de Tecnología de Georgia con el objetivo de ofrecer a cualquier usuario acceso a una plataforma donde poder realizar experimentos con robots reales de forma remota y libre [4].

Este laboratorio (Figura 2.2a) alberga una flota de pequeños robots móviles diferenciales¹ de tipo uniciclo, denominados GRITSbots (Figura 2.2b). Estos robots cuentan con dos ruedas laterales y una esfera delantera, lo cual les permite girar sobre su propio eje, pero tienen restringido el desplazamiento lateral. Por tanto, son robots no holonómicos².



(a) Plataforma robótica para pruebas experimentales



(b) GRITSbot X

Figura 2.2: Robotarium, ubicado en las instalaciones del *Georgia Tech* [4].

El movimiento de estos robots es analizado por las ocho cámaras de alta precisión instaladas en el techo de la sala, que obtienen información sobre la posición de cada agente. El usuario remoto puede ejecutar un algoritmo de control basado en estas posiciones. Además, durante la ejecución, el sistema es capaz de predecir futuras colisiones y corregir el rumbo automáticamente. Esto se consigue gracias a las *Control Barrier Functions* (CBF) que incorpora Robotarium en su *software*, evitando que los robots choquen entre ellos o con los límites del banco de pruebas donde se realiza el experimento.

¹Se entiende por vehículo diferencial aquel en el que cada rueda puede girar en un sentido y velocidad diferentes.

²De manera simplificada, se considera que un robot es no holonómico cuando no puede cambiar su dirección de manera inmediata y, por tanto, precisa realizar maniobras previas de desplazamiento y/o rotación.

Capítulo 3

Control multi-robot con modos de deformación en entornos 2D

Se propone un controlador cuyo objetivo es dirigir un equipo de múltiples robots móviles, gobernados por dinámicas de integrador simple, utilizando modos de deformación que permitan controlar, en todo momento, la deformación que pueda sufrir la formación durante su trayectoria.

Este control se ha diseñado considerando un grupo de N robots encargados de transportar un objeto deformable en un espacio bidimensional hasta una configuración deseada, resultante de la combinación de cuatro parámetros: forma, escala, posición y orientación. Se ha supuesto que los robots están unidos al objeto mediante articulaciones rotacionales, por lo que el giro de los mismos sobre su propio eje no afecta a la deformación del elemento transportado.

Puesto que se asumen dinámicas de integrador simple, cada robot $i \in \{1, \dots, N\}$ se mueve según $\dot{\mathbf{p}}_i = \mathbf{u}_i$, donde $\mathbf{p}_i = [p_{ix}, p_{iy}]^\top$ denota la posición del agente y $\mathbf{u}_i = [u_{ix}, u_{iy}]^\top$ es la velocidad calculada por el controlador. Agrupar posiciones y velocidades según $\mathbf{p} = [\mathbf{p}_1^\top, \dots, \mathbf{p}_N^\top]^\top \in \mathbb{R}^{2N}$ y $\mathbf{u} = [\mathbf{u}_1^\top, \dots, \mathbf{u}_N^\top]^\top \in \mathbb{R}^{2N}$, respectivamente, facilita definir el control de todo el equipo de robots al mismo tiempo.

La estrategia de control, desarrollada a lo largo de los siguientes apartados, se plantea como una combinación lineal de los controladores individuales de los cuatro parámetros que definen la configuración de la formación (forma, escala, posición y orientación). Nótese que el control de la forma incluye, por un lado, el control de la conservación de forma y, por otro, el control de las deformaciones sufridas. Aunque ambos influyen en la forma del equipo de robots, se controlan de manera independiente.

Así, la ley de control completa, \mathbf{u} , se define como sigue:

$$\mathbf{u} = \mathbf{u}_H + \mathbf{u}_G + \mathbf{u}_s + \mathbf{u}_c + \mathbf{u}_\theta , \quad (3.1)$$

donde \mathbf{u}_H es el control de conservación de forma, \mathbf{u}_G es el control de deformaciones, \mathbf{u}_s es el control de escala, \mathbf{u}_c es el control de posición y \mathbf{u}_θ es el control de orientación.

3.1. Control de la deformación

Una característica de los objetos deformables es su capacidad para adaptarse a la forma del equipo de robots que los transporta. Por ello, su deformación se puede restringir controlando directamente la forma y la escala de la formación robótica.

3.1.1. Control de forma

El control de forma se realiza respecto a una configuración de referencia (Figura 3.1a) definida por $\mathbf{c} = [\mathbf{c}_1^\top, \dots, \mathbf{c}_N^\top]^\top \in \mathbb{R}^{2N}$, donde $\mathbf{c}_i = [c_{ix}, c_{iy}]^\top$ es la posición del robot i . Esto es así porque se considera que la forma de la configuración deseada es igual a la de la configuración de referencia, independientemente de la escala, posición y orientación.

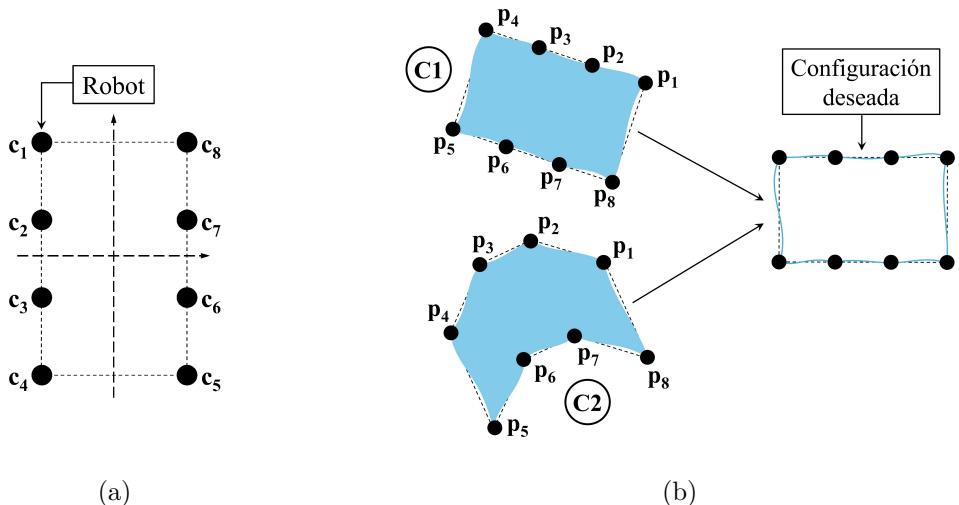


Figura 3.1: (a) Configuración de referencia. (b) Representación de dos configuraciones alcanzables por el equipo de robots durante el transporte de un objeto deformable al implementar el controlador con modos de deformación propuesto.

La formación se encuentra en una **configuración de conservación de forma** (Configuración C1 en la Figura 3.1b) si las posiciones \mathbf{p}_i resultan de aplicar una traslación, rotación y escalado a las posiciones \mathbf{c}_i . Por conveniencia y practicidad, el componente de traslación se trata moviendo los dos conjuntos de posiciones para hacer su centroide cero. Concretamente, definiendo $\mathbf{p}'_i = \mathbf{p}_i - \mathbf{g}$ y $\mathbf{c}'_i = \mathbf{c}_i - \mathbf{g}_c$, donde \mathbf{g} y \mathbf{g}_c representan los centroides de los grupos de puntos \mathbf{p}_i y \mathbf{c}_i , la condición a cumplir es

$$\mathbf{p}'_i = \mathbf{H}\mathbf{c}'_i \quad \text{para todo } i , \quad (3.2)$$

donde la matriz \mathbf{H} representa un escalado y una rotación, y se define como

$$\mathbf{H} = \begin{bmatrix} h_1 & -h_2 \\ h_2 & h_1 \end{bmatrix} \in \mathbb{R}^{2 \times 2} . \quad (3.3)$$

La condición anterior se puede expresar de manera conjunta para todos los robots como

$$\begin{bmatrix} p_{1x} & \dots & p_{Nx} \\ p_{1y} & \dots & p_{Ny} \end{bmatrix} \cdot \mathbf{K}_N = \mathbf{H} \cdot \begin{bmatrix} c_{1x} & \dots & c_{Nx} \\ c_{1y} & \dots & c_{Ny} \end{bmatrix} \cdot \mathbf{K}_N , \quad (3.4)$$

donde $\mathbf{K}_N = (\mathbf{I}_N - (1/N)\mathbf{1}_N\mathbf{1}_N^\top)$ es una matriz, denominada *centering matrix*, que traslada el centroide de la configuración al origen de coordenadas.

En otras palabras, para considerar que se está en una configuración de conservación de forma, la formación robótica debe tener la misma forma que la configuración de referencia. Definiendo la siguiente matriz, donde $\mathbf{K}_{2N} = \mathbf{K}_N \otimes \mathbf{I}_2 \in \mathbb{R}^{2N \times 2N}$:

$$\mathbf{C}_H = \mathbf{K}_{2N} \cdot \begin{bmatrix} c_{1x} & c_{1y} & \dots & c_{Nx} & c_{Ny} \\ -c_{1y} & c_{1x} & \dots & -c_{Ny} & c_{Nx} \end{bmatrix}^\top \in \mathbb{R}^{2N \times 2} , \quad (3.5)$$

la condición de la Ecuación 3.4 equivale a $\mathbf{K}_{2N}\mathbf{p} - \mathbf{C}_H\mathbf{h}_H = \mathbf{0}$, con $\mathbf{h}_H = [h_1, h_2]^\top \in \mathbb{R}^2$.

Se busca definir un término de control para lograr que las posiciones \mathbf{p}_i cumplan la condición de conservación de forma. Para este fin, en primer lugar se escoge el valor de \mathbf{h}_H usando la técnica de los mínimos cuadrados, de manera que $\|\mathbf{K}_{2N}\mathbf{p} - \mathbf{C}_H\mathbf{h}_H\|$ sea mínimo. Se puede demostrar que $\mathbf{C}_H^+\mathbf{K}_{2N} = \mathbf{C}_H^+$ y, por lo tanto, se tiene que $\mathbf{h}_H = \mathbf{C}_H^+\mathbf{K}_{2N}\mathbf{p} = \mathbf{C}_H^+\mathbf{p}$. Con esto, se define la función de coste asociada con la conservación de forma:

$$\gamma_H = \frac{1}{2}\|\mathbf{K}_{2N}\mathbf{p} - \mathbf{C}_H\mathbf{h}_H\|^2 = \frac{1}{2}\mathbf{p}^\top \mathbf{A}_H \mathbf{p} , \quad (3.6)$$

donde $\mathbf{A}_H = \mathbf{K}_{2N} - \mathbf{C}_H\mathbf{C}_H^+$ es una matriz constante, simétrica¹, idempotente² y semidefinida positiva³.

¹Una matriz es simétrica si es cuadrada e igual a su traspuesta.

²Una matriz \mathbf{A} es idempotente si $\mathbf{A}^2 = \mathbf{A}$.

³Una matriz simétrica $\mathbf{A} \in \mathbb{R}^{n \times n}$ es semidefinida positiva si $\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0$ para todo $\mathbf{x} \in \mathbb{R}^n$ no nulo.

Así, siendo k_H una ganancia de control positiva, el término de control relativo a la conservación de forma del equipo de robots se define siguiendo el gradiente negativo de γ_H :

$$\mathbf{u}_H = -k_H \mathbf{A}_H \mathbf{p} . \quad (3.7)$$

Por otro lado, de manera similar, el equipo de robots se encuentra en una **configuración deformada expresada por modos de deformación de orden dos** (Configuración C2 en la Figura 3.1b) si

$$\begin{bmatrix} p_{1x} & \dots & p_{Nx} \\ p_{1y} & \dots & p_{Ny} \end{bmatrix} \cdot \mathbf{K}_N = \mathbf{G} \cdot \begin{bmatrix} c_{1x} & \dots & c_{Nx} \\ c_{1y} & \dots & c_{Ny} \\ c_{1x}^2 & \dots & c_{Nx}^2 \\ c_{1y}^2 & \dots & c_{Ny}^2 \\ c_{1x} c_{1y} & \dots & c_{Nx} c_{Ny} \end{bmatrix} \cdot \mathbf{K}_N , \quad (3.8)$$

donde \mathbf{G} es una matriz que engloba los parámetros lineales (l_i), cuadráticos (q_i) y mixtos (m_i) encargados de generar una configuración deformada respecto de la configuración de referencia. Esta matriz se define como

$$\mathbf{G} = \begin{bmatrix} l_1 & l_2 & q_1 & q_2 & m_1 \\ l_3 & l_4 & q_3 & q_4 & m_2 \end{bmatrix} \in \mathbb{R}^{2 \times 5} . \quad (3.9)$$

Si la formación robótica se encuentra en este tipo de configuración, la deformación sufrida estará causada por estiramiento (*stretch*), cizalla (*shear*), doblado (*bend*), torsión (*twist*) o cualquier combinación de los mismos. Los dos primeros modos de deformación se representan mediante términos lineales, mientras que los dos últimos se describen usando términos cuadráticos. En la Figura 3.2 se ejemplifican cuatro posibles deformaciones que puede sufrir la configuración rectangular de referencia considerada.

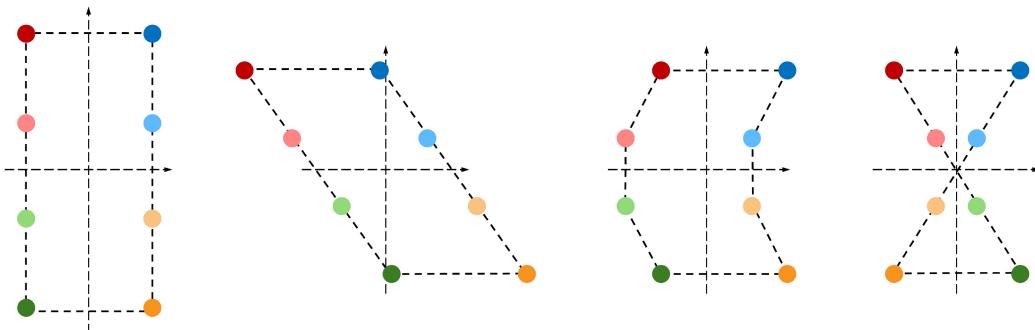


Figura 3.2: Modos de deformación. De izquierda a derecha: *stretch*, *shear*, *bend* y *twist*.

Para ver más ejemplos de deformaciones alcanzables generadas por cada término individual de la Ecuación 3.9 acudir al Anexo B.

Inspirado en el trabajo de Müller *et al.* [16], se define la matriz

$$\mathbf{C}_G = \mathbf{K}_{2N} \cdot \begin{bmatrix} \mathbf{L}_1 & \mathbf{Q}_1 & \mathbf{M}_1 \\ \vdots & \vdots & \vdots \\ \mathbf{L}_N & \mathbf{Q}_N & \mathbf{M}_N \end{bmatrix} \in \mathbb{R}^{2N \times 10}, \quad (3.10)$$

donde los términos lineales (\mathbf{L}_i), cuadráticos (\mathbf{Q}_i) y mixtos (\mathbf{M}_i) se calculan como:

$$\mathbf{L}_i = \begin{bmatrix} c_{ix} & c_{iy} & 0 & 0 \\ 0 & 0 & c_{ix} & c_{iy} \end{bmatrix} \in \mathbb{R}^{2 \times 4}, \quad (3.11)$$

$$\mathbf{Q}_i = \begin{bmatrix} c_{ix}^2 & c_{iy}^2 & 0 & 0 \\ 0 & 0 & c_{ix}^2 & c_{iy}^2 \end{bmatrix} \in \mathbb{R}^{2 \times 4}, \quad (3.12)$$

$$\mathbf{M}_i = \begin{bmatrix} c_{ix}c_{iy} & 0 \\ 0 & c_{ix}c_{iy} \end{bmatrix} \in \mathbb{R}^{2 \times 2}. \quad (3.13)$$

De forma similar al control de conservación de forma, se busca un vector $\mathbf{h}_G \in \mathbb{R}^{10}$ que minimice $\|\mathbf{K}_{2N}\mathbf{p} - \mathbf{C}_G\mathbf{h}_G\|$. Tras calcular la función de coste

$$\gamma_G = \frac{1}{2}\|\mathbf{K}_{2N}\mathbf{p} - \mathbf{C}_G\mathbf{h}_G\|^2 = \frac{1}{2}\mathbf{p}^\top \mathbf{A}_G \mathbf{p}, \quad (3.14)$$

se propone el término de control asociado a la deformación del equipo de robots:

$$\mathbf{u}_G = -k_G \mathbf{A}_G \mathbf{p}, \quad (3.15)$$

donde k_G es una ganancia de control positiva y $\mathbf{A}_G = \mathbf{K}_{2N} - \mathbf{C}_G \mathbf{C}_G^+$ es una matriz constante, simétrica, idempotente y semidefinida positiva.

En la Figura 3.3 se expone el resultado de aplicar, por separado, cada uno de los controladores mencionados en este apartado. Con esto, se pretende mostrar cómo el controlador \mathbf{u}_H provoca que la formación alcance la forma de la configuración de referencia, mientras que el controlador \mathbf{u}_G dirige a los robots hacia una configuración deformada cuadráticamente. El efecto de usar ambos controles combinados, $\mathbf{u}_H + \mathbf{u}_G$, se refleja en la Figura 3.4a.

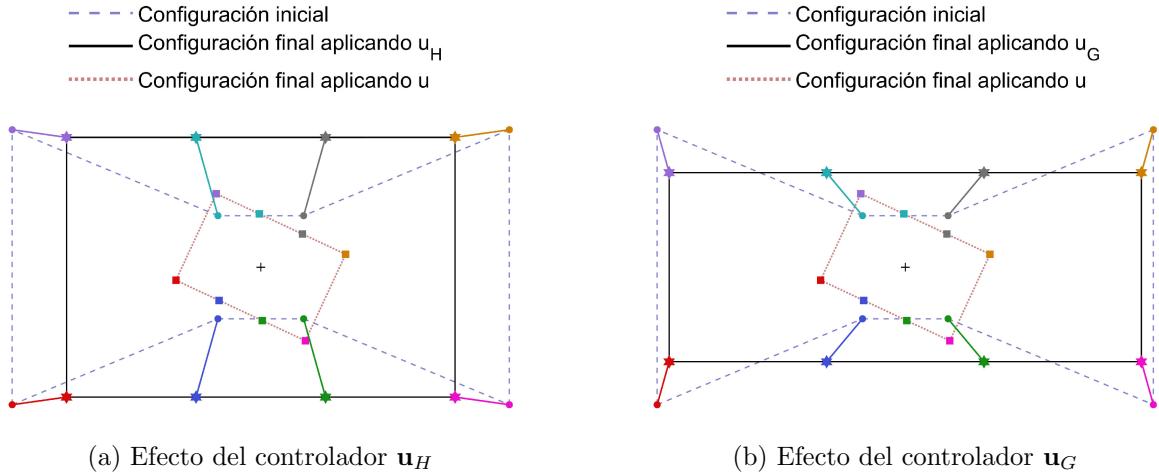


Figura 3.3: Trayectorias de los robots al aplicar el control de conservación de forma, \mathbf{u}_H , y el control de deformación, \mathbf{u}_G , individualmente. El equipo de robots se encuentra inicialmente en las posiciones marcadas por círculos unidos con línea azul discontinua y, al aplicar los controladores, alcanzan la configuración representada por estrellas unidas con línea negra continua. Los cuadrados conectados por líneas rojas punteadas indican el estado al que llegaría la formación robótica si se aplicase el control completo, \mathbf{u} , que corresponde con la configuración deseada.

3.1.2. Control de escala

Para controlar por completo la deformación del equipo de robots es necesario controlar también la escala, definida como $s = \|\mathbf{h}_H\|$. El siguiente término de control garantiza que la formación tienda hacia la escala deseada, s_d :

$$\mathbf{u}_s = k_s (s_d - s) (1/s) \mathbf{C}_H \mathbf{h}_H , \quad (3.16)$$

donde k_s es una ganancia de control positiva. El resultado de controlar únicamente la escala de la formación puede verse en la Figura 3.4b. En este caso, se ha definido la configuración inicial más pequeña que la configuración deseada para una mejor visualización.

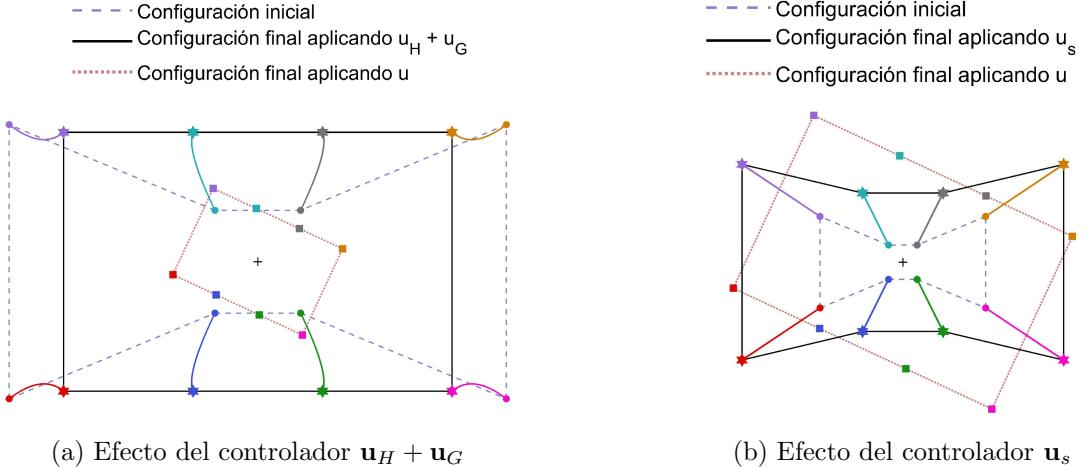


Figura 3.4: Trayectorias de los robots al aplicar el control de forma, $\mathbf{u}_H + \mathbf{u}_G$, y el control de escala, \mathbf{u}_s , individualmente. El equipo de robots se encuentra inicialmente en las posiciones marcadas por círculos unidos con línea azul discontinua y, al aplicar los controladores, alcanzan la configuración representada por estrellas unidas con línea negra continua. Los cuadrados conectados por líneas rojas punteadas indican el estado al que llegaría la formación robótica si se aplicase el control completo, \mathbf{u} , que corresponde con la configuración deseada.

3.2. Control de traslación y control de rotación

La traslación y la rotación del equipo no afectan a la deformación del objeto, ya que no modifican las posiciones relativas de los robots. El control de traslación se encarga de dirigir a la formación robótica en conjunto de forma que su centroide, definido como $\mathbf{g} = \frac{1}{N} [\mathbf{p}_1, \dots, \mathbf{p}_N] \mathbf{1}_N$, alcance la posición del centroide deseado, \mathbf{g}_d . Por otro lado, el control de orientación rota la forma del equipo en torno a su centroide hasta que el ángulo en el que está orientado, $\theta = \text{atan2}(h_2, h_1)$, coincide con el ángulo deseado, θ_d . En la práctica, se puede tomar $\theta_d = 0$ por conveniencia.

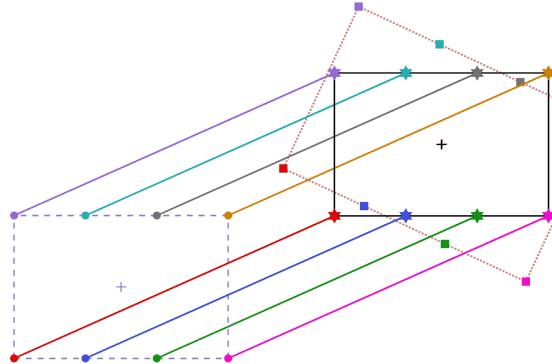
A continuación, se proponen los términos de control asociados a la posición y la orientación, respectivamente:

$$\mathbf{u}_c = k_c \mathbf{1}_N \otimes (\mathbf{g}_d - \mathbf{g}) , \quad (3.17)$$

$$\mathbf{u}_\theta = k_\theta (\theta_d - \theta) (\mathbf{I}_N \otimes \mathbf{S}) \mathbf{C}_H \mathbf{h}_H , \quad (3.18)$$

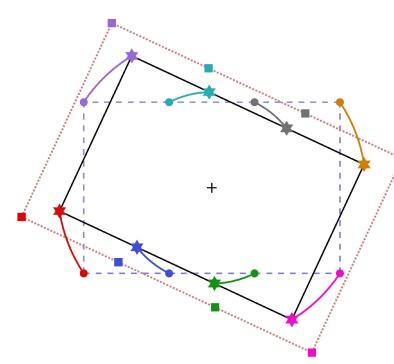
donde k_c y k_θ son ganancias de control positivas. El resultado de implementar los controles de posición y de orientación de forma individual se refleja en la Figura 3.5.

— Configuración inicial
— Configuración final aplicando \mathbf{u}_c
... Configuración final aplicando \mathbf{u}



(a) Efecto del controlador \mathbf{u}_c

— Configuración inicial
— Configuración final aplicando \mathbf{u}_θ
... Configuración final aplicando \mathbf{u}



(b) Efecto del controlador \mathbf{u}_θ

Figura 3.5: Trayectorias de los robots al aplicar el control de posición, \mathbf{u}_c , y el control de orientación, \mathbf{u}_θ , individualmente. El equipo de robots se encuentra inicialmente en las posiciones marcadas por círculos unidos con línea azul discontinua y, al aplicar los controladores, alcanzan la configuración representada por estrellas unidas con línea negra continua. Los cuadrados conectados por líneas rojas punteadas indican el estado al que llegaría la formación robótica si se aplicase el control completo, \mathbf{u} , que corresponde con la configuración deseada.

3.3. Métricas de error

La evolución de los errores de los cuatro parámetros considerados durante el desplazamiento del equipo de robots permite validar el correcto funcionamiento del controlador propuesto. El error de forma se evalúa mediante las funciones de coste γ_H (Ecuación 3.6) y γ_G (Ecuación 3.14). Los errores de posición (e_g), escala (e_s) y orientación (e_θ) se definen a continuación:

$$e_g = \|\mathbf{g} - \mathbf{g}_d\| , \quad (3.19)$$

$$e_s = s - s_d , \quad (3.20)$$

$$e_\theta = |\theta - \theta_d| . \quad (3.21)$$

En el capítulo siguiente, se utilizarán estas variables de error para analizar la efectividad del controlador completo en diferentes escenarios, tanto simulados como reales.

Capítulo 4

Evaluación experimental en entornos 2D

Para verificar el correcto funcionamiento del controlador, se plantean diferentes configuraciones y escenarios en los que el equipo de robots debe transportar un objeto deformable hasta una configuración final deseada.

La evaluación experimental del algoritmo de control propuesto se ha llevado a cabo mediante diversas simulaciones en el simulador de MATLAB proporcionado por Robotarium, así como de forma remota en su banco de pruebas con robots reales. Estos experimentos se describen y analizan detalladamente en los siguientes apartados. Además, se dedica una sección para explicar el proceso de implementación del controlador en Robotarium y la problemática encontrada durante la misma. Salvo que se indique lo contrario, los objetos deformables en los ejemplos mostrados en esta memoria se simulan utilizando el método *As-Rigid-As-Possible* (ARAP) [23], bajo la premisa de que el objeto se deforme lo mínimo necesario.

El Anexo C recoge resultados gráficos adicionales de los experimentos mostrados a continuación, así como algún ensayo complementario.

4.1. Implementación en Robotarium

Utilizar Robotarium aporta muchas ventajas a la hora de ajustar y analizar el controlador desarrollado, pero también posee ciertas limitaciones y requisitos a tener en cuenta, como no poder realizar los experimentos con objetos reales.

Un aspecto muy importante a tener en cuenta es el relativo al movimiento de los GRITSbots, los cuales se desplazan en base a órdenes de velocidad. El controlador implementado devuelve el parámetro de velocidad como un vector cartesiano (u_x, u_y), puesto que utiliza dinámicas de integrador simple. Por el contrario, los robots, al ser uniciclos, solo son capaces de interpretar los comandos en términos de velocidad lineal (v) y velocidad angular (ω). La adaptación de velocidades holonómicas a velocidades de uniciclo se consigue mediante una serie de transformaciones geométricas que implican la proyección de la posición del robot una distancia l en la dirección de movimiento.

El hecho de tener que aplicar esta distancia de proyección supone que el controlador trabaje con una posición «ficticia» de los robots, lo que genera un error final en la posición y una rotación constante de los robots sobre su propio eje cuando están próximos a la posición deseada. El primer problema se soluciona proyectando la posición final del centroide una distancia igual a la distancia de proyección, cuyo valor óptimo se ha considerado como $l = 5 \text{ cm}$. Para el segundo, basta con imponer una condición de parada, de manera que cada robot se detenga cuando esté a menos de 2 cm de su posición final.

Otro requisito imprescindible es el uso de las CBF para evitar cualquier posible colisión. Sin ellas, Robotarium no permite ejecutar ningún experimento en sus instalaciones. Estas funciones modifican la dirección de la velocidad en caso de que los robots se encuentren a menos de cierta distancia de seguridad de un obstáculo o de otro robot. También pueden reducir el módulo de la velocidad si el valor calculado por el controlador excede el máximo permitido, que en todos los experimentos realizados en Robotarium a lo largo de este trabajo se ha asumido como $v_{max} = 0.2 \text{ m/s}$.

La Figura 4.1 resume, en forma de diagrama de bloques, la implementación del bucle de control desarrollado.



Figura 4.1: Diagrama de bloques del bucle de control utilizado, donde γ_d representa la forma deseada y γ la forma de la formación en cada iteración.

El código desarrollado para implementar el controlador propuesto en Robotarium se muestra en el Anexo E. Para más información sobre el proceso de implantación del algoritmo de control en la plataforma Robotarium, consultar [3].

4.2. Transporte de objeto deformable en entorno 2D sin obstáculos

La parte más novedosa del controlador implementado frente al de trabajos anteriores es la incorporación del control de deformaciones, restringiendo el cambio de forma del equipo de robots durante toda la tarea de transporte. Para analizar el efecto que tiene este controlador individual en el conjunto de la formación se considera un experimento, evaluado mediante simulación en MATLAB, en el que 12 robots se encuentran, inicialmente, en una configuración puramente doblada y son dirigidos hacia una configuración final rectangular. A continuación, se plantean dos situaciones.

Caso 1. Se controlan todos los parámetros de la formación, excepto el relativo a las deformaciones cuadráticas. Para ello, se escogen los siguientes valores de ganancias de control: $k_H = 5$, $k_G = 0$, $k_s = 2$, $k_c = 0.5$ y $k_\theta = 0.15$. Como se puede ver en la Figura 4.2a, el equipo de robots alcanza la configuración deseada, pero el objeto sufre deformaciones durante el transporte que podrían dañarlo.

Caso 2. Se controlan todos los parámetros de la formación, esta vez tomando $k_G = 10$. El resto de ganancias de control tienen el mismo valor que en el Caso 1. Aquí, las deformaciones del objeto están controladas y su integridad física está garantizada. En la Figura 4.2b se puede apreciar que el objeto transportado tiende a mantener una configuración cuadrática *bend*.

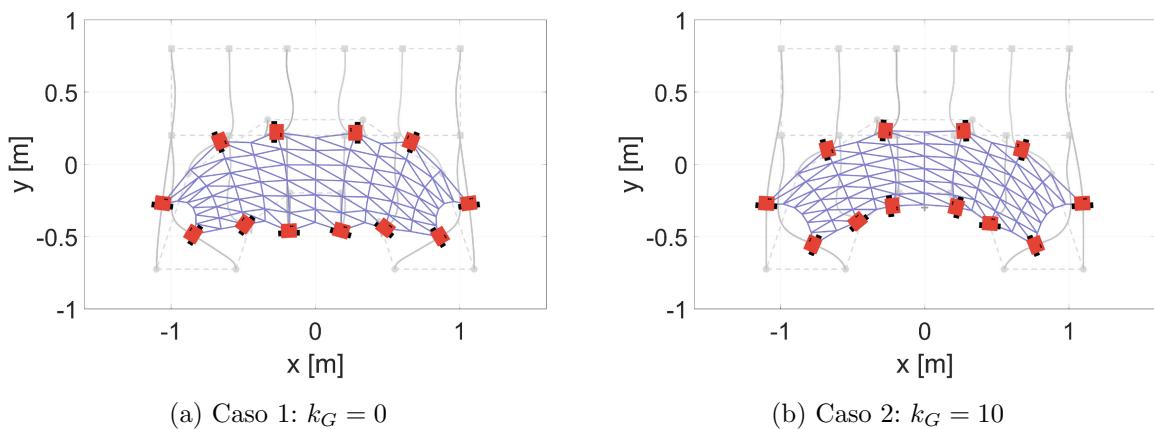


Figura 4.2: Comparación de la configuración de la formación en el instante $t = 0.8$ s en los dos casos considerados del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular.

Desde el Anexo F.1 se puede acceder al vídeo de esta simulación, donde se comparan las deformaciones sufridas a lo largo del transporte en ambos casos.

Aunque la diferencia de forma durante el transporte es el cambio más notable, el efecto de aplicar el controlador \mathbf{u}_G también se refleja en las gráficas de evolución de los errores (Figura 4.3). En el Caso 1, la variable γ_G alcanza valores cercanos a cero prácticamente al mismo tiempo que γ_H . Recordando lo explicado en el Capítulo 3, la función de coste γ_G representa la diferencia de forma entre la configuración de referencia deformada según modos de orden dos y la configuración de la formación en un instante dado. Por tanto, parece lógico que el equipo de robots solo alcance una configuración cuadrática al mismo tiempo que llega a la configuración de conservación de forma, modelada por \mathbf{u}_H . Esto es así porque se considera que la configuración de conservación de forma pertenece al conjunto de posibles configuraciones deformadas. Es decir, $\gamma_H = 0$ implica que $\gamma_G = 0$. Y, en ese caso, $\mathbf{G} = [\mathbf{H}, \mathbf{0}]$.

Por el contrario, puesto que en el Caso 2 sí se controlan las deformaciones durante el transporte, γ_G converge antes que γ_H . Es decir, la formación alcanza una configuración deformada cuadráticamente (en este caso, una configuración *bend*) antes de llegar a la configuración deseada. El resto de variables de error se ven ligeramente afectadas como consecuencia de las nuevas maniobras que ejecutan los robots, pero no se percibe ningún cambio significativo.

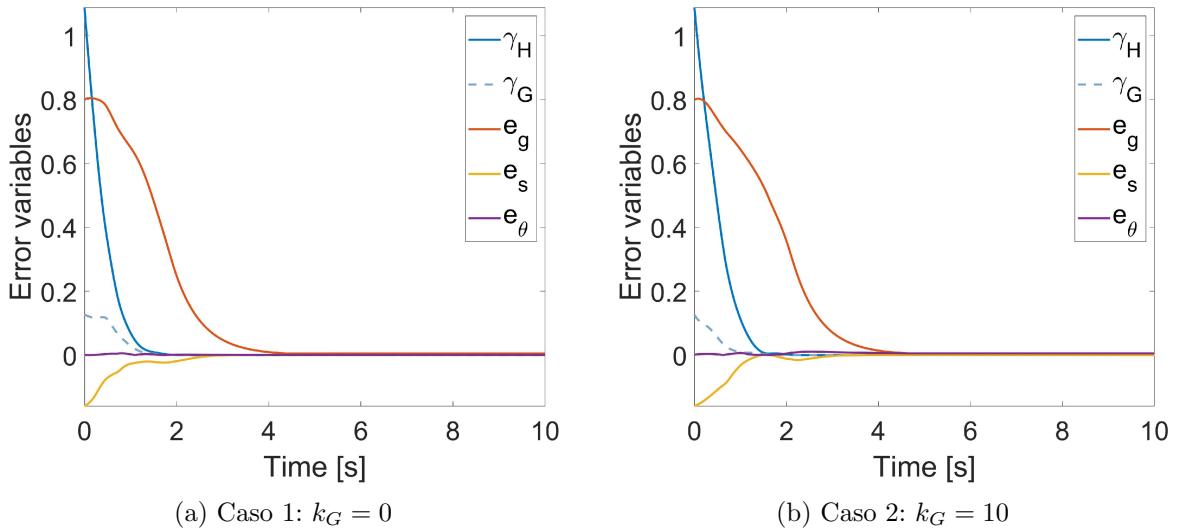


Figura 4.3: Comparación de la evolución de las variables de error en los dos casos considerados del experimento simulado donde el equipo de robots pasa de una configuración puramente dobrada a una configuración rectangular.

El resto de resultados obtenidos en esta simulación se recogen en el Anexo C.1.

4.3. Transporte de objeto deformable en entorno 2D con obstáculo

Uno puede pensar que la simulación anterior se trata de un caso particular donde el control \mathbf{u}_G ha cumplido su función por estar la configuración inicial ya deformada. Para confirmar que el controlador implementado funciona correctamente cuando las deformaciones deben realizarse en mitad de la tarea de transporte, se propone una simulación en un espacio bidimensional con obstáculos.

El objetivo es dirigir un equipo de 8 robots desde una configuración inicial rectangular hasta una configuración deseada de la misma forma y escala, pero rotada 90°. En mitad del área de movimiento, se ha ubicado un obstáculo circular que la formación debe esquivar para llegar a la posición deseada garantizando, en todo momento, que el objeto transportado no sufra ningún daño. Se plantean dos situaciones.

Caso 1. Se escogen las siguientes ganancias de control: $k_H = 0.2$, $k_G = 0$, $k_s = 0.15$, $k_c = 0.1$ y $k_\theta = 0.15$. Es decir, las deformaciones que surjan durante el transporte no estarán controladas. En la Figura 4.4a, se percibe claramente cómo el equipo de robots está deformando el objeto de manera extraña para poder esquivar el obstáculo. Si se tratase de un objeto real, esta deformación podría dañarlo o, incluso, llegar a romperlo.

Caso 2. El único cambio en las ganancias respecto al Caso 1 es que ahora $k_G = 2$, por lo que el equipo se deformará de manera controlada. Como se muestra en la Figura 4.4b, la deformación que sufre el objeto es mucho más natural.

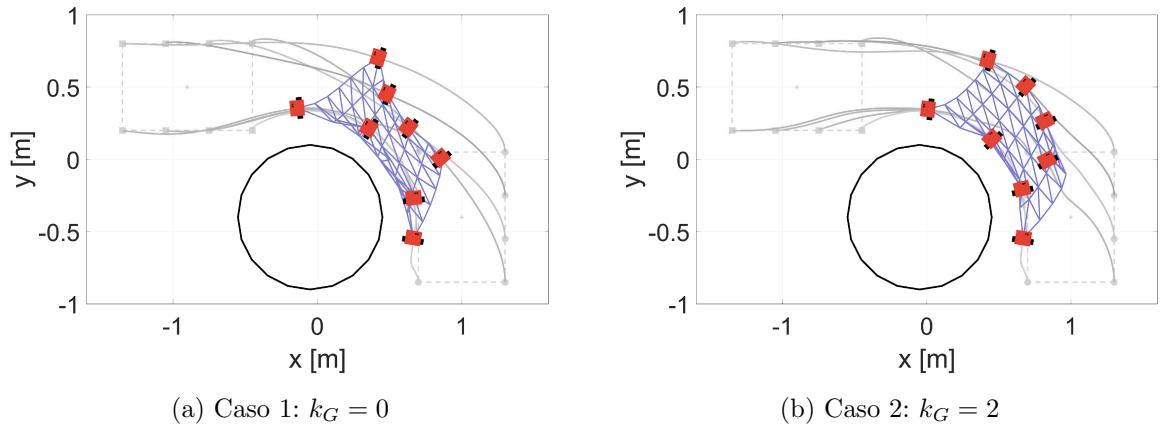


Figura 4.4: Comparación de la configuración de la formación en el instante $t = 2$ s en los dos casos considerados del experimento simulado donde el equipo de robots debe esquivar un obstáculo circular.

Desde el Anexo F.2 se puede acceder al vídeo donde se comparan ambos casos.

La evolución de las variables de error (Figura 4.5) también permite apreciar la diferencia entre aplicar el control de deformaciones o no. Es importante darse cuenta de que los valores de γ_H y γ_G graficados son un orden de magnitud superiores a los reales para facilitar la comparación entre casos.

En el Caso 1, tanto γ_H como γ_G sufren una fuerte variación durante el tramo de simulación en el que la formación robótica está esquivando el obstáculo. Aun así, γ_G es, aproximadamente, cuatro veces inferior a γ_H . Esto se debe a que, durante la deformación, el equipo de robots está más próximo a una configuración deformada cuadráticamente que a la configuración de conservación de forma.

En el Caso 2, se percibe una reducción de los valores de γ_H y γ_G , especialmente de esta última variable. Al controlar cómo se puede deformar la formación robótica, γ_G permanece en valores prácticamente nulos durante toda la simulación. Esto hace que, a su vez, el error de forma γ_H también se vea reducido.

Analizando el resto de variables se observa que casi no se ven afectadas por la aplicación del controlador \mathbf{u}_G , pero se distinguen unos pequeños «rebotes» en e_s y e_θ . En el caso de e_s , se debe a la propia deformación, que obliga al equipo de robots a modificar su escala. Si mantener la escala del objeto fuese de gran importancia, se podría dar más peso a \mathbf{u}_s , ya que este controlador está desacoplado del resto. En cuanto a e_θ , el cambio brusco es consecuencia de calcularlo como valor absoluto, aunque la evolución del ángulo θ a lo largo del tiempo es suave. Para más resultados de esta simulación, así como un ensayo adicional, acudir al Anexo C.2.

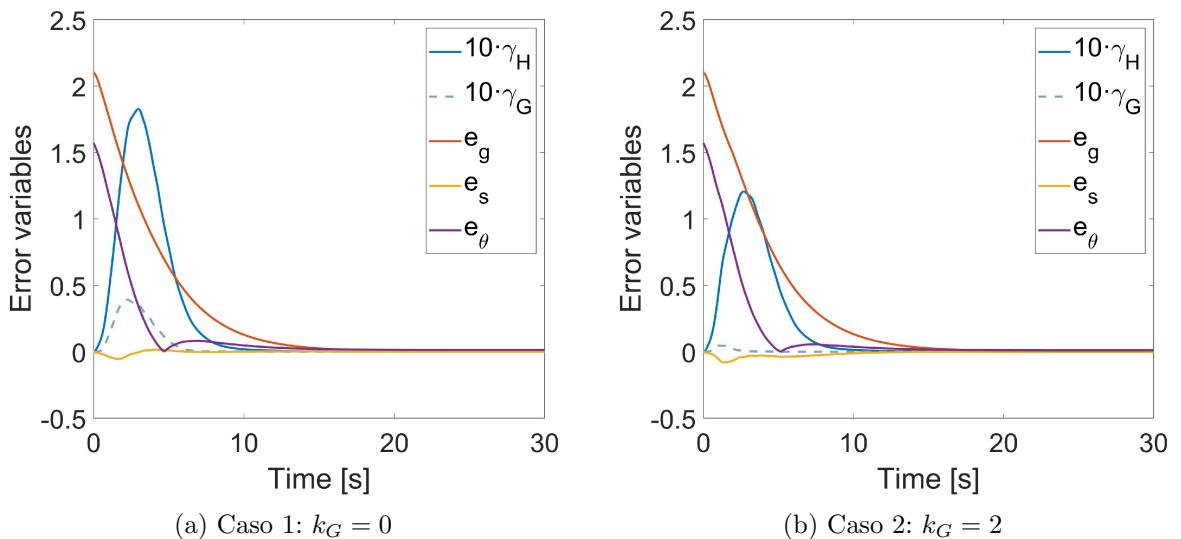


Figura 4.5: Comparación de la evolución de las variables de error en los dos casos considerados del experimento simulado donde el equipo de robots debe esquivar un obstáculo circular.

4.4. Transporte de objeto deformable en entorno 2D por pasillo curvo

Los resultados obtenidos mediante simulación ofrecen una primera aproximación del comportamiento del sistema, pero no tienen en cuenta factores externos que pueden afectar al movimiento de los robots. Para realizar una evaluación más completa del controlador y ratificar su correcto funcionamiento, resulta fundamental su implementación y análisis con Robotarium.

Se propone un experimento en el que un equipo de 8 robots reales debe transportar un objeto deformable a través de un pasillo curvo. En la Figura 4.6 se muestra el escenario donde tiene lugar este experimento.

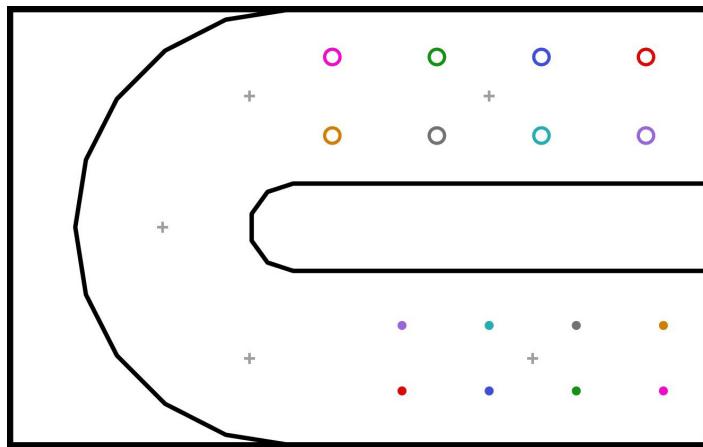


Figura 4.6: Representación del escenario en el que se va a llevar a cabo el experimento donde el equipo de robots recorre un pasillo curvo en un espacio 2D. Los puntos de colores son las posiciones iniciales de los robots y los círculos huecos representan las posiciones finales deseadas.

Habitualmente, el control de robots móviles en entornos complejos suele requerir la implementación de algoritmos de planificación de rutas que indiquen a los robots el camino a seguir. Este es uno de esos casos que precisan de una planificación previa de la ruta. Los robots, al no conocer a priori la existencia de los obstáculos, tienden a moverse directamente hacia la parte superior, donde se localiza la configuración final que deben alcanzar. Al hacer esto, llegan a un estado de bloqueo.

Por ello, se ha decidido introducir tres puntos intermedios, o *waypoints*, a lo largo del pasillo, representados por una cruz gris en la figura anterior. El controlador \mathbf{u}_c se encarga de dirigir al equipo hacia el siguiente *waypoint* del recorrido, así hasta llegar a la posición final. Estos *waypoints* están definidos de forma manual como la combinación de la forma, escala, posición y orientación que debe tener la formación en este momento.

Se plantean tres situaciones para estudiar este experimento, dos de ellas bajo las mismas condiciones de control que las simulaciones de los apartados anteriores.

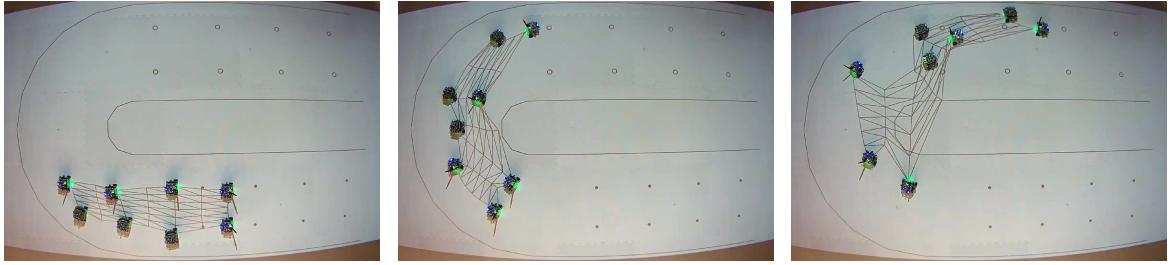
Caso 0. Solo se controlan tres de los cuatro parámetros que definen la configuración: posición, orientación y escala. La forma del equipo de robots no se restringe. Los valores elegidos para las ganancias de control son $k_H = 0$, $k_G = 0$, $k_s = 0.1$, $k_c = 0.1$ y $k_\theta = 0.1$. En la Figura 4.7a, se puede ver, en los momentos más avanzados del experimento, cómo el objeto está totalmente deformado. Al no estar controlada la conservación de forma durante el transporte, cualquier maniobra no coordinada o bloqueo provoca una excesiva deformación.

Caso 1. La estrategia aplicada en este caso consiste en controlar la conservación de forma del objeto, pero no las deformaciones. Por eso, se escogen $k_H = 0.2$ y $k_G = 0$. El resto de ganancias son iguales al caso anterior. La Figura 4.7b muestra cómo el equipo de robots trata de mantener la forma original del objeto, pero se ve obligado a deformarse al evitar chocar con las paredes del pasillo. Estas deformaciones no están controladas, por lo que el objeto adquiere una forma extraña.

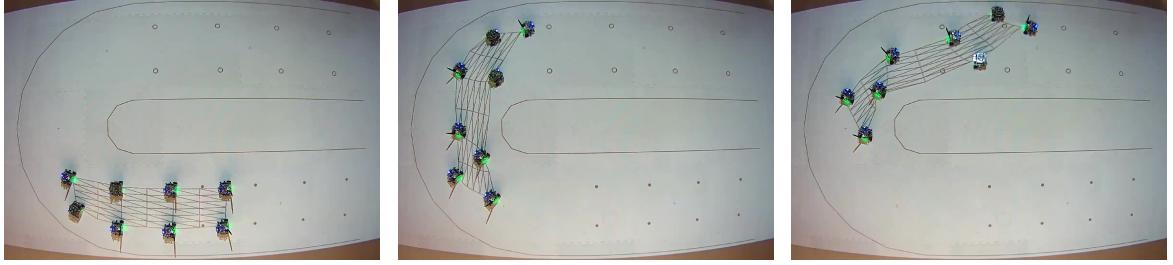
Caso 2. Se controlan todos los parámetros de la configuración, incluidas la conservación de forma y las deformaciones. Ahora, se escoge $k_G = 1$, mientras que el resto de ganancias tienen el mismo valor que en el Caso 1. En la Figura 4.7c, se observa que, aunque la formación robótica deba deformarse, lo hace cuadráticamente.

El conjunto de fotogramas seleccionado muestra la gran utilidad del controlador propuesto. En espacios libres donde el equipo de robots no tiene que deformarse, como el primer tramo recto del pasillo, cualquiera de las estrategias seguidas en los Casos 1 o 2 daría buenos resultados. Con la estrategia del Caso 0, cada robot iría directo a su posición final sin tener en cuenta la integridad del objeto transportado.

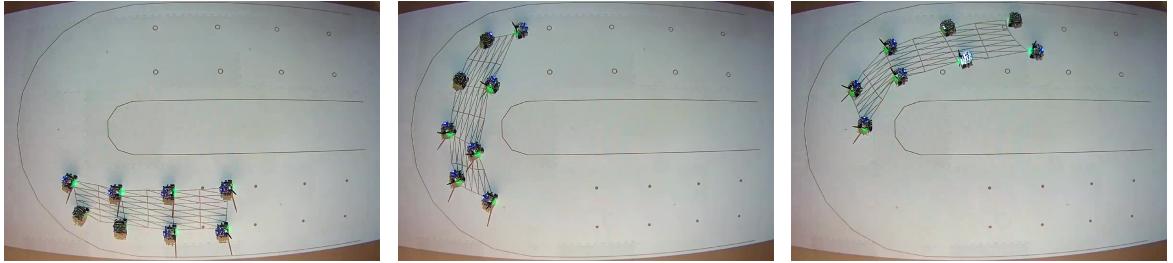
En cambio, cuando la formación es forzada a deformarse, como al doblar la esquina del pasillo, claramente el control de forma con modos de deformación del Caso 2 es una mejor opción. Con él, el equipo de robots realiza movimientos de forma coordinada teniendo en cuenta, en todo momento, las propiedades del objeto transportado para deformarlo de manera controlada.



(a) Caso 0: $k_H = 0$ y $k_G = 0$



(b) Caso 1: $k_H = 0.2$ y $k_G = 0$



(c) Caso 2: $k_H = 0.2$ y $k_G = 1$

Figura 4.7: Comparación de la configuración de la formación en los instantes $t = 15$ s (izquierda), $t = 35$ s (centro) y $t = 50$ s (derecha) en los tres casos considerados del experimento donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

En el Anexo F.3 se muestran los enlaces para acceder a los vídeos donde se comparan los tres casos del experimento en simulación y con robots reales.

La evolución de los errores durante la tarea (Figura 4.8) también refleja la diferencia entre las tres estrategias. Por un lado, en el Caso 0, la tendencia de γ_H y γ_G es creciente. Es decir, al no estar controlando la forma, el controlador nunca corregirá este error y su valor solo podrá aumentar. En este caso particular, γ_H y γ_G toman valores tan elevados debido a que uno de los robots se queda bloqueado en la parte inferior del pasillo.

En cuanto a los Casos 1 y 2, se puede observar que γ_H y γ_G sí convergen a cero, aunque con alguna oscilación intermedia, más suaves en el segundo de estos casos por estar las deformaciones controladas. Además, en el tramo temporal comprendido entre 50 y 100 segundos, la variable γ_H permanece constante y positiva, consecuencia del bloqueo momentáneo de un robot.

Por otro lado, se puede observar que los errores de posición y orientación casi no varían entre un caso y otro, mientras que el de escala sí se ve afectado en cierta medida por la deformación sufrida. Nótese que los valores de γ_H y γ_G han sido multiplicados por 10 en las gráficas de los Casos 1 y 2.

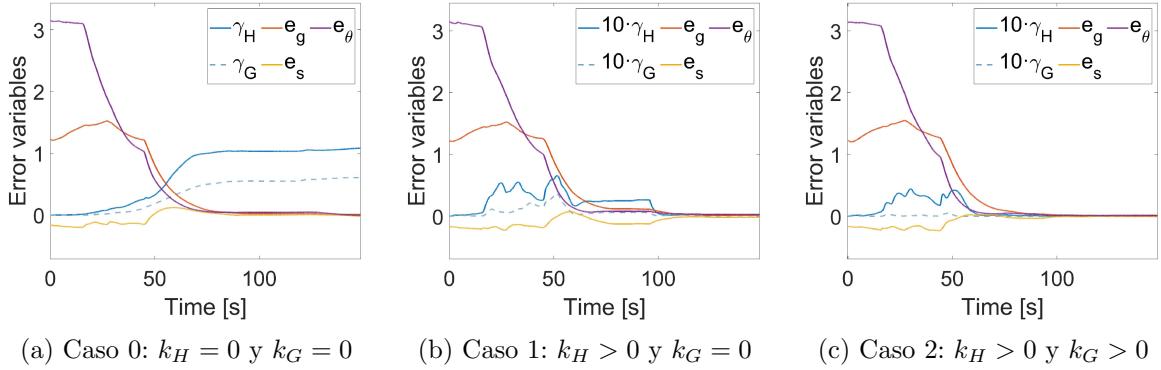


Figura 4.8: Comparación de la evolución de las variables de error en los tres casos considerados del experimento donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

Otro parámetro interesante es la velocidad lineal de los uniciclos (Figura 4.9), cuyo mejor resultado se obtiene en el Caso 2. Los cambios bruscos observados en las gráficas se deben a la presencia de los *waypoints*, ya que la formación reduce su velocidad al aproximarse a uno de ellos y, cuando se encuentra a cierta distancia, pasa a dirigirse hacia el siguiente *waypoint*, aumentando considerablemente la velocidad.

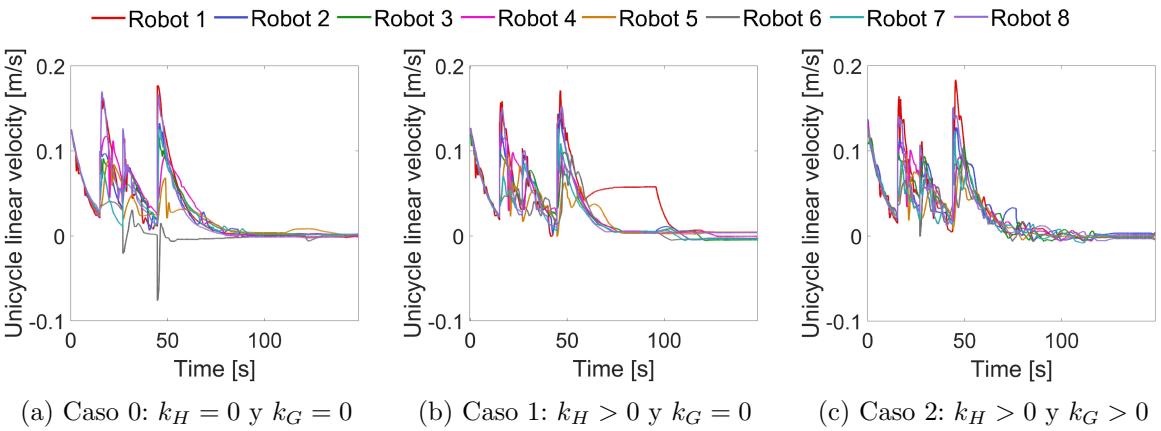


Figura 4.9: Comparación de la variación de velocidad lineal de cada robot en los tres casos considerados del experimento donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

Con todo esto, viendo que la deformación que sufre el objeto durante su transporte sigue patrones cuadráticos y que todas las variables de error convergen a cero, se puede concluir que el controlador implementado cumple los objetivos propuestos. En el Anexo C.3 se muestran más gráficas de este experimento, tanto de simulación como de los ensayos con robots reales.

Capítulo 5

Extensión del controlador a entornos 3D

La manipulación de objetos deformables no solo es una tarea relevante en espacios bidimensionales. El constante desarrollo tecnológico y el auge de la robótica aérea hacen que trasportar estos objetos en entornos 3D resulte un tema de gran interés en el ámbito de la investigación. Por ello, se pretende adaptar el controlador propuesto en el Capítulo 3 para dirigir un equipo de robots en entornos tridimensionales. Debido a la posibilidad de rotación en las tres dimensiones del espacio, el control de este parámetro es más complejo y requiere la aplicación de métodos adicionales, como la aproximación mediante rotaciones infinitesimales o el uso del algoritmo de Kabsch.

Se siguen asumiendo dinámicas de integrador simple, por lo que el comando de movimiento de cada robot $i \in \{1, \dots, N\}$ también viene dado por $\dot{\mathbf{p}}_i = \mathbf{u}_i$. Las posiciones y velocidades se definen como $\mathbf{p}_i = [p_{ix}, p_{iy}, p_{iz}]^\top$ y $\mathbf{u}_i = [u_{ix}, u_{iy}, u_{iz}]^\top$, agrupadas en $\mathbf{p} = [\mathbf{p}_1^\top, \dots, \mathbf{p}_N^\top]^\top \in \mathbb{R}^{3N}$ y $\mathbf{u} = [\mathbf{u}_1^\top, \dots, \mathbf{u}_N^\top]^\top \in \mathbb{R}^{3N}$. La configuración de referencia se denota por $\mathbf{c} = [\mathbf{c}_1^\top, \dots, \mathbf{c}_N^\top]^\top \in \mathbb{R}^{3N}$, con $\mathbf{c}_i = [c_{ix}, c_{iy}, c_{iz}]^\top$.

La estrategia de control se mantiene, combinar linealmente los controles individuales de forma $(\mathbf{u}_H + \mathbf{u}_G)$, escala (\mathbf{u}_s) , posición (\mathbf{u}_c) y rotación $(\mathbf{u}_R$ según el Apartado 5.1 o \mathbf{u}_{Hd} según el Apartado 5.2) en un único controlador (\mathbf{u}) . La ley de control implementada cuando se usan rotaciones infinitesimales es, por tanto,

$$\mathbf{u} = \mathbf{u}_H + \mathbf{u}_G + \mathbf{u}_s + \mathbf{u}_c + \mathbf{u}_R , \quad (5.1)$$

mientras que, al utilizar el algoritmo de Kabsch, la ley de control se describe como

$$\mathbf{u} = \mathbf{u}_H + \mathbf{u}_G + \mathbf{u}_s + \mathbf{u}_c + \mathbf{u}_{Hd} . \quad (5.2)$$

5.1. Control con rotaciones infinitesimales

Una rotación se considera infinitesimal cuando el ángulo rotado en torno a un eje es muy pequeño, permitiendo simplificar en gran medida la representación de la matriz de rotación. Por tanto, ofrece resultados aproximados, no exactos. En particular, en el contexto de este trabajo, la representación de las rotaciones como infinitesimales permite evitar el uso de algoritmos numéricos, como el de Kabsch, en la ley de control.

5.1.1. Control de forma

Configuración de conservación de forma. Igual que en el caso 2D, el equipo de robots se encuentra en esta configuración si $\mathbf{p}_i = \mathbf{H}\mathbf{c}_i$, con centroide en $(0, 0, 0)$. Inspirado en [19], se considera una matriz \mathbf{H} de alineación óptima cuya estructura representa un escalado uniforme y rotaciones infinitesimales¹ respecto de la configuración de referencia:

$$\mathbf{H} = \begin{bmatrix} h_s & -h_{rz} & h_{ry} \\ h_{rz} & h_s & -h_{rx} \\ -h_{ry} & h_{rx} & h_s \end{bmatrix} \in \mathbb{R}^{3 \times 3}, \quad (5.3)$$

cuyos parámetros pueden agruparse en un único vector $\mathbf{h}_H = [h_s, \mathbf{h}_r^\top]^\top \in \mathbb{R}^{4 \times 1}$, siendo $\mathbf{h}_r = [h_{rx}, h_{ry}, h_{rz}]^\top$.

La condición anterior equivale a $\mathbf{K}_{3N}\mathbf{p} - \mathbf{C}_H\mathbf{h}_H = \mathbf{0}$, donde $\mathbf{K}_{3N} = \mathbf{K}_N \otimes \mathbf{I}_3 \in \mathbb{R}^{3N \times 3N}$. Ahora, la matriz \mathbf{C}_H se define como sigue:

$$\mathbf{C}_H = \mathbf{K}_{3N} \cdot \begin{bmatrix} c_{1x} & c_{1y} & c_{1z} & \dots & c_{Nx} & c_{Ny} & c_{Nz} \\ 0 & -c_{1z} & c_{1y} & \dots & 0 & -c_{Nz} & c_{Ny} \\ c_{1z} & 0 & -c_{1x} & \dots & c_{Nz} & 0 & -c_{Nx} \\ -c_{1y} & c_{1x} & 0 & \dots & -c_{Ny} & c_{Nx} & 0 \end{bmatrix}^\top \in \mathbb{R}^{3N \times 4}. \quad (5.4)$$

Mediante la técnica de los mínimos cuadrados, se busca el vector \mathbf{h}_H óptimo que minimice $\|\mathbf{K}_{3N}\mathbf{p} - \mathbf{C}_H\mathbf{h}_H\|$. Esto se puede resolver como $\mathbf{h}_H = \mathbf{C}_H^+ \mathbf{K}_{3N}\mathbf{p} = \mathbf{C}_H^+ \mathbf{p} = [h_s, h_{rx}, h_{ry}, h_{rz}]^\top$.

¹Esta representación de \mathbf{H} asume una estructura de la matriz de rotación como la mostrada en la Ecuación 5.11. Es comúnmente usada para representar rotaciones con ángulos infinitesimales y resulta de tomar una aproximación de primer orden del seno y el coseno del ángulo de la rotación. Por tanto, representa una aproximación válida cuando el ángulo de rotación es pequeño. A lo largo de la memoria, se denomina «rotaciones infinitesimales» a todas aquellas que poseen esta estructura.

Así, la función de coste asociada a la conservación de forma es

$$\gamma_H = \frac{1}{2} \|\mathbf{K}_{3N} \mathbf{p} - \mathbf{C}_H \mathbf{h}_H\|^2 = \frac{1}{2} \mathbf{p}^\top \mathbf{A}_H \mathbf{p}, \quad (5.5)$$

donde $\mathbf{A}_H = \mathbf{K}_{3N} - \mathbf{C}_H \mathbf{C}_H^+$ sigue siendo una matriz constante, simétrica, idempotente y semidefinida positiva.

El término de control que modela la conservación de forma se define igual que en la Ecuación 3.7.

Configuración deformada según modos de orden dos. El equipo se encuentra en esta configuración si $\mathbf{p}_i = \mathbf{G} \cdot [c_{ix}, c_{iy}, c_{iz}, c_{ix}^2, c_{iy}^2, c_{iz}^2, c_{ix}c_{iy}, c_{iy}c_{iz}, c_{iz}c_{ix}]^\top$ para todos los robots i , de nuevo tomando el centroide en $(0, 0, 0)$, y siendo

$$\mathbf{G} = \begin{bmatrix} l_1 & l_2 & l_3 & q_1 & q_2 & q_3 & m_1 & m_2 & m_3 \\ l_4 & l_5 & l_6 & q_4 & q_5 & q_6 & m_4 & m_5 & m_6 \\ l_7 & l_8 & l_9 & q_7 & q_8 & q_9 & m_7 & m_8 & m_9 \end{bmatrix} \in \mathbb{R}^{3 \times 9}. \quad (5.6)$$

En este caso, la matriz \mathbf{C}_G se define como

$$\mathbf{C}_G = \mathbf{K}_{3N} \cdot \begin{bmatrix} \mathbf{L}_1 & \mathbf{Q}_1 & \mathbf{M}_1 \\ \vdots & \vdots & \vdots \\ \mathbf{L}_N & \mathbf{Q}_N & \mathbf{M}_N \end{bmatrix} \in \mathbb{R}^{3N \times 27}, \quad (5.7)$$

donde, ahora, los términos lineales ($\mathbf{L}_i \in \mathbb{R}^{3 \times 9}$), cuadráticos ($\mathbf{Q}_i \in \mathbb{R}^{3 \times 9}$) y mixtos ($\mathbf{M}_i \in \mathbb{R}^{3 \times 9}$) son:

$$\mathbf{L}_i = \begin{bmatrix} c_{ix} & c_{iy} & c_{iz} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{ix} & c_{iy} & c_{iz} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{ix} & c_{iy} & c_{iz} \end{bmatrix}, \quad (5.8)$$

$$\mathbf{Q}_i = \begin{bmatrix} c_{ix}^2 & c_{iy}^2 & c_{iz}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{ix}^2 & c_{iy}^2 & c_{iz}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{ix}^2 & c_{iy}^2 & c_{iz}^2 \end{bmatrix}, \quad (5.9)$$

$$\mathbf{M}_i = \begin{bmatrix} c_{ix}c_{iy} & c_{iy}c_{iz} & c_{iz}c_{ix} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{ix}c_{iy} & c_{iy}c_{iz} & c_{iz}c_{ix} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{ix}c_{iy} & c_{iy}c_{iz} & c_{iz}c_{ix} \end{bmatrix}. \quad (5.10)$$

La función de coste γ_G y el control de deformaciones \mathbf{u}_G se expresan igual que en las Ecuaciones 3.14 y 3.15, respectivamente.

5.1.2. Control de escala, posición y orientación

El control de escala es igual a la Ecuación 3.16, pero con $s = h_s$. Lo mismo ocurre con el término de control de posición, idéntico a la Ecuación 3.17, con $\mathbf{g} \in \mathbb{R}^3$.

En cambio, el control de orientación se ve ligeramente modificado, debido a la posibilidad de rotación en los tres ejes del espacio. Como se ha mencionado antes, la matriz de transformación \mathbf{H} es una combinación de escalado y rotación, por lo que su valor óptimo se puede reformular como $\mathbf{H} = h_s \mathbf{R}$, donde \mathbf{R} es la matriz de rotaciones infinitesimales, definida como sigue:

$$\mathbf{R} = \begin{bmatrix} 1 & -h_{rz}/h_s & h_{ry}/h_s \\ h_{rz}/h_s & 1 & -h_{rx}/h_s \\ -h_{ry}/h_s & h_{rx}/h_s & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 3}. \quad (5.11)$$

Esta matriz de rotación también se puede escribir como $\mathbf{R} = \mathbf{I}_3 + \mathbf{S}_r$, siendo \mathbf{S}_r una matriz antisimétrica² formulada, a partir de los elementos del vector \mathbf{h}_H , como $\mathbf{S}_r = \left[\frac{1}{h_s} \mathbf{h}_r \right]_\times$. Nótese, además, que $\mathbf{R}^{-1} \approx \mathbf{R}^\top = \mathbf{I}_3 - \mathbf{S}_r$.

El control de orientación tiene como objetivo rotar la formación alrededor del centroide para llegar a una rotación deseada \mathbf{R}_d . Dándose cuenta de que \mathbf{S}_r es equivalente a la matriz antisimétrica \mathbf{S} de la Ecuación 3.18, se puede definir el término de control de orientación de igual forma que en 2D:

$$\mathbf{u}_R = k_R (\mathbf{I}_N \otimes (\mathbf{R}_d - \mathbf{R})) \mathbf{C}_H \mathbf{h}_H = -k_R (\mathbf{I}_N \otimes \mathbf{S}_r) \mathbf{C}_H \mathbf{h}_H, \quad (5.12)$$

donde k_R es una ganancia de control positiva y $\mathbf{R}_d = \mathbf{I}_3$ por simplicidad y por consistencia con la aproximación (\mathbf{R}) de la rotación, la cual es más precisa cuanto más cercana es la rotación real a \mathbf{I}_3 . La ventaja del método descrito en este apartado es que ofrece una definición analítica del controlador.

5.2. Control con algoritmo de Kabsch

El algoritmo de Kabsch es una técnica matemática utilizada para calcular la matriz de rotación óptima que minimice la diferencia entre dos conjuntos de puntos. Esta matriz influye no solo en el control de orientación, sino también en el control de forma y el control de escala, tal como se explica en los siguientes apartados.

²Una matriz \mathbf{A} se considera antisimétrica si cumple que $\mathbf{A}^\top = -\mathbf{A}$.

El método seguido para calcular la matriz de rotación se basa en el trabajo de Sorkine y Rabinovich [20]. A partir de las matrices de posición, con centroide en $(0, 0, 0)$, en la configuración de referencia $\mathbf{C}_b = [\mathbf{c}_1 \dots \mathbf{c}_N] \cdot \mathbf{K}_N \in \mathbb{R}^{3 \times N}$ y en la configuración actual $\mathbf{P}_b = [\mathbf{p}_1 \dots \mathbf{p}_N] \cdot \mathbf{K}_N \in \mathbb{R}^{3 \times N}$, se calcula la matriz de covarianza (\mathbf{M}_K) según

$$\mathbf{M}_K = \mathbf{C}_b \mathbf{W} \mathbf{P}_b , \quad (5.13)$$

donde \mathbf{W} es una matriz de pesos que, en este caso, toma el valor $\mathbf{W} = \mathbf{I}_N$. El subíndice K indica que la variable correspondiente se ha calculado siguiendo el algoritmo de Kabsch, facilitando así la diferenciación con las variables del apartado anterior.

Tras aplicar la descomposición en valores singulares (SVD) a esta matriz de covarianza, tal que $\mathbf{M}_K = \mathbf{U}_K \mathbf{S}_K \mathbf{V}_K^\top$, se obtiene la matriz de rotación como sigue:

$$\mathbf{R}_K = \mathbf{V}_K \cdot \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & & & \det(\mathbf{V}_K \mathbf{U}_K^\top) \end{bmatrix} \cdot \mathbf{U}_K . \quad (5.14)$$

5.2.1. Control de forma

Configuración de conservación de forma. Se toma como referencia el método descrito en Aranda *et al.* [21], el cual fue también usado en el trabajo [22]. Se define la matriz de transformación $\mathbf{H}_K = s_K \mathbf{R}_K$, donde la escala se calcula como $s_K = \text{tr}(\mathbf{P}_b^\top \mathbf{R}_K \mathbf{C}_b) / c_s$, con $c_s = \text{tr}(\mathbf{C}_b \mathbf{C}_b^\top) = \|\mathbf{C}_b\|_F^2$.

La función de coste que permite evaluar la disparidad de forma se formula, ahora, como

$$\gamma_H = \frac{1}{2} \|\mathbf{P}_b - \mathbf{H}_K \mathbf{C}_b\|_F^2 . \quad (5.15)$$

Así, el término de control asociado a la conservación de forma queda definido como

$$\mathbf{u}_H = k_H \cdot \text{vec}(\mathbf{H}_K \mathbf{C}_b - \mathbf{P}_b) . \quad (5.16)$$

Este control de forma implica el uso de un algoritmo numérico en cada instante de tiempo durante la ejecución. Por su parte, el control de forma con rotaciones infinitesimales, el cual se implementa de acuerdo a la Ecuación 3.7, tiene una formulación analítica y que depende linealmente de las posiciones de los robots siendo, por tanto, más fácil de tratar.

Configuración deformada según modos de orden dos. El control de las deformaciones durante la tarea de transporte, así como el cálculo de su función de coste, se realizan exactamente igual que lo explicado en el Apartado 5.1.

5.2.2. Control de escala, posición y orientación

El control de posición es idéntico a la Ecuación 3.17, con $\mathbf{g} \in \mathbb{R}^3$, mientras que el término de control de escala se reformula de la siguiente forma:

$$\mathbf{u}_s = k_s \cdot \text{vec}((s_d - s_K)(1/s_K) \mathbf{H}_K \mathbf{C}_b) . \quad (5.17)$$

En cuanto a la rotación, se propone un término de control que engloba tanto la orientación como la escala (\mathbf{u}_{Hd}). El objetivo es que el equipo de robots se mueva hacia una transformación final deseada $\mathbf{H}_{Kd} = s_d \mathbf{R}_d$, aplicando escalado y rotación. Como es posible controlar la escala de la formación de manera independiente con el controlador de la Ecuación 5.17, el siguiente término de control es una manera efectiva de controlar la orientación:

$$\mathbf{u}_{Hd} = k_{Hd} \cdot \text{vec}(\mathbf{H}_{Kd} \mathbf{C}_b - \mathbf{P}_b) , \quad (5.18)$$

donde k_{Hd} es una ganancia de control positiva.

5.3. Métricas de error en entornos 3D

Igual que en 2D, el error de forma se evalúa a partir de las funciones de coste γ_H y γ_G . La primera de estas variables se calcula según las Ecuaciones 5.5 y 5.15, dependiendo de si se implementa el control con rotaciones infinitesimales o con el algoritmo de Kabsch, respectivamente. La variable γ_G se expresa según la Ecuación 3.14 en ambos casos.

Asimismo, los errores de posición y escala se definen, respectivamente, mediante la Ecuación 3.19, con $\mathbf{g} \in \mathbb{R}^3$, y la Ecuación 3.20, con $s = s_K$ en caso de aplicar el algoritmo de Kabsch.

El error en la orientación (e_R) se obtiene a partir de la matriz de rotación. Sabiendo que una matriz multiplicada por su inversa es la matriz identidad, con rotaciones infinitesimales se tiene que

$$e_R = \|\mathbf{I}_3 - \mathbf{R} \mathbf{R}_d^{-1}\|_F , \quad (5.19)$$

mientras que con el algoritmo de Kabsch se calcula como

$$e_R = \|\mathbf{I}_3 - \mathbf{R}_K \mathbf{R}_d^{-1}\|_F . \quad (5.20)$$

Además, resulta interesante comparar el error de orientación en las tres dimensiones de espacio respecto del ángulo deseado $\theta_d = 0$. Estos errores se calculan en base a los ejes x ($e_{\theta x}$), y ($e_{\theta y}$) y z ($e_{\theta z}$) de la referencia local de la configuración deseada, siendo

$$e_{\theta x} = |\theta_x - \theta_d| , \quad (5.21)$$

$$e_{\theta y} = |\theta_y - \theta_d| , \quad (5.22)$$

$$e_{\theta z} = |\theta_z - \theta_d| , \quad (5.23)$$

con θ_x , θ_y y θ_z los ángulos de Euler en el eje local correspondiente.

Capítulo 6

Evaluación experimental en entornos 3D

Ya comprobado el correcto funcionamiento en entornos 2D, se debe proceder de igual manera para verificar que el controlador propuesto también permite dirigir un equipo de robots en un espacio tridimensional. Por ello, se plantean diversos ejemplos en los que una formación robótica transporta un objeto deformable desde una configuración inicial hasta una configuración final. En todos los experimentos realizados se ha considerado que el objeto transportado es superficial, excepto en el Apartado 6.3, donde se muestra una simulación con un objeto volumétrico.

A diferencia del caso 2D, solo se han podido realizar los ensayos en simulación, donde los robots se han modelado mediante dinámicas de integrador simple. Además, los resultados permitirán comparar el control de rotación mediante el algoritmo de Kabsch y las rotaciones infinitesimales en términos de precisión y estabilidad y, así, determinar el más adecuado para el control de equipos multi-robot en entornos 3D.

Todos los resultados mostrados a continuación pertenecen a experimentos donde se controlan todos los parámetros de la formación, como son la conservación de forma, las deformaciones sufridas durante el transporte, la escala, la posición y la orientación. En el Anexo D se recogen resultados adicionales, así como experimentos complementarios.

6.1. Transporte de objeto deformable en entorno 3D sin rotación

Se propone un experimento en el que un equipo de 14 robots se encuentra, inicialmente, en una configuración puramente doblada y se dirigen hacia una configuración final rectangular en el espacio 3D. En este primer ejemplo no existe rotación relativa entre ambas configuraciones, con el fin de facilitar el análisis del efecto de los controladores en la deformación del objeto. Los valores escogidos para las ganancias de control son $k_H = 1$, $k_G = 5$, $k_s = 0.5$, $k_c = 1$ y $k_R = k_{Hd} = 0.8$.

La Figura 6.1 demuestra que la deformación sufrida, tanto al aplicar el control con rotaciones infinitesimales como con el algoritmo de Kabsch, es muy similar. La principal diferencia es el tamaño de la configuración en el instante representado. Esto puede verse también en la Figura 6.2, donde el error de escala con rotaciones infinitesimales es más lento que con el algoritmo de Kabsch, tanto que no termina de converger en el tiempo de simulación fijado. Para tiempos de simulación mayores, el error de escala alcanza valores nulos. Por otro lado, el error de forma γ_H varía ligeramente, reduciéndose más suavemente en el caso de las rotaciones infinitesimales.

Los errores de posición (e_g) y deformación (γ_G) no se ven modificados y, puesto que se calculan de forma idéntica en ambos casos, esto es un indicativo del desacople entre los dos conjuntos de controladores.

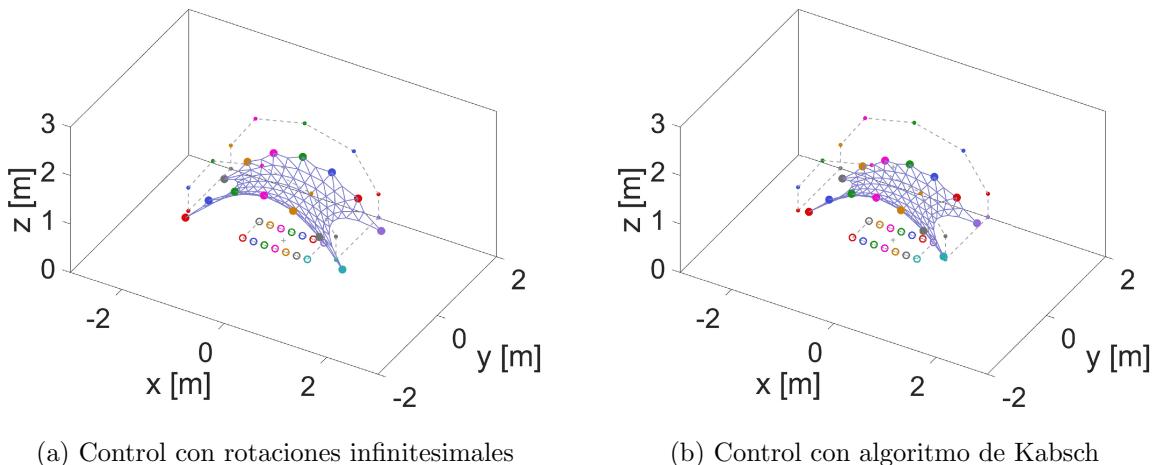
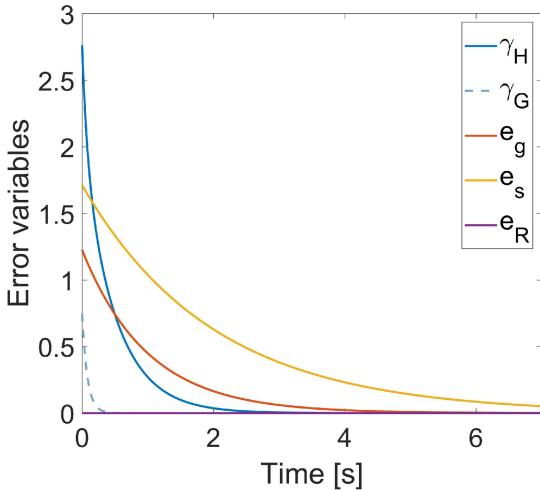
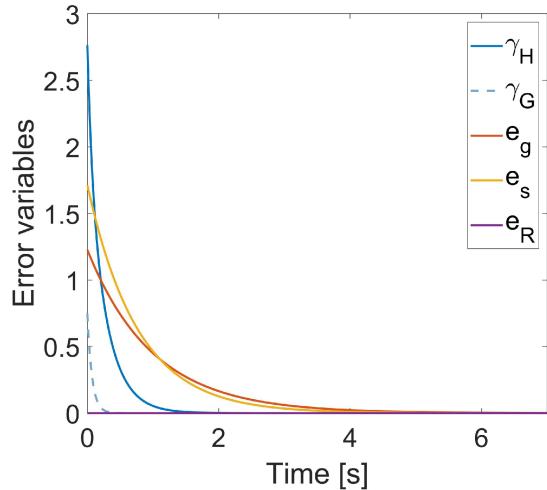


Figura 6.1: Comparación de la configuración de la formación en el instante $t = 0.4$ s al aplicar los dos métodos de control considerados, con $k_G = 5$, en el experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D.



(a) Control con rotaciones infinitesimales

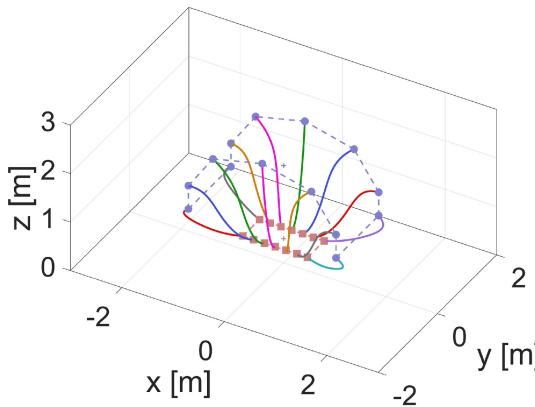


(b) Control con algoritmo de Kabsch

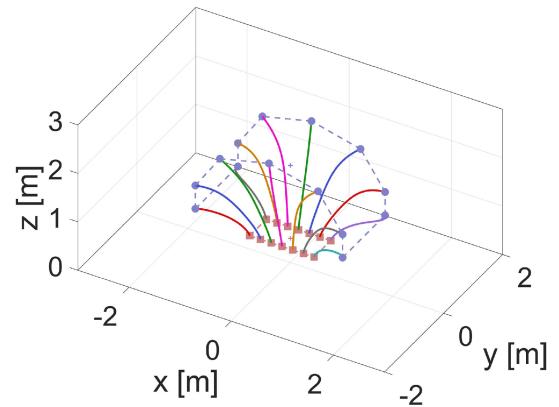
Figura 6.2: Comparación de la evolución de las variables de error al aplicar los dos métodos de control considerados, con $k_G = 5$, en el experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D.

Las trayectorias de la Figura 6.3 muestran las diferencias en el movimiento de la formación robótica. Dado que la matriz de rotación es siempre igual a \mathbf{I}_3 en ambos casos, las diferencias observadas son principalmente debidas a que en el control con el algoritmo de Kabsch hay dos términos (Ecuaciones 5.17 y 5.18) que corrigen conjuntamente (y por tanto, más rápidamente) el error e_s .

● Posiciones iniciales — Robot 1 — Robot 3 — Robot 5 — Robot 7 — Robot 9 — Robot 11 — Robot 13
 ■ Posiciones deseadas — Robot 2 — Robot 4 — Robot 6 — Robot 8 — Robot 10 — Robot 12 — Robot 14



(a) Control con rotaciones infinitesimales



(b) Control con algoritmo de Kabsch

Figura 6.3: Comparación de las trayectorias seguidas por los robots al aplicar los dos métodos de control considerados, con $k_G = 5$, en el experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D.

El Anexo D.1 completa estos resultados con gráficas que permiten comparar el efecto de implementar o no el controlador \mathbf{u}_G . El vídeo donde se puede ver la deformación sufrida por el objeto durante el transporte está accesible a través del Anexo F.5.

6.2. Transporte de objeto deformable en entorno 3D con rotación

Para poder realizar un estudio completo del funcionamiento del controlador debe haber cierta rotación entre la configuración inicial y la final. Por eso, en el siguiente experimento se dirige un equipo de 8 robots desde una configuración inicial deformada en las tres dimensiones del espacio hasta una configuración deseada rectangular y con una orientación diferente. Las ganancias de control toman los siguientes valores: $k_H = 1$, $k_G = 2$, $k_s = 0.5$, $k_c = 0.5$ y $k_R = k_{Hd} = 0.8$. En la Figura 6.4 se puede ver el escenario descrito, así como la deformación del objeto en el instante $t = 0.8$ s, diferente según el control implementado. Se puede acceder al vídeo a través del Anexo F.6.

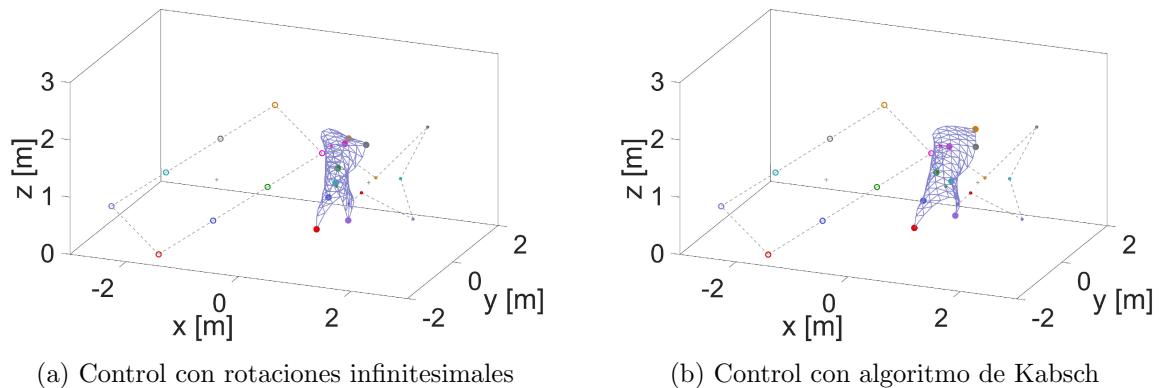
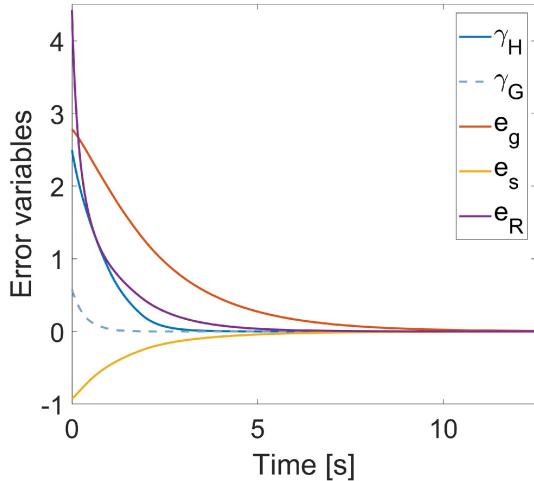
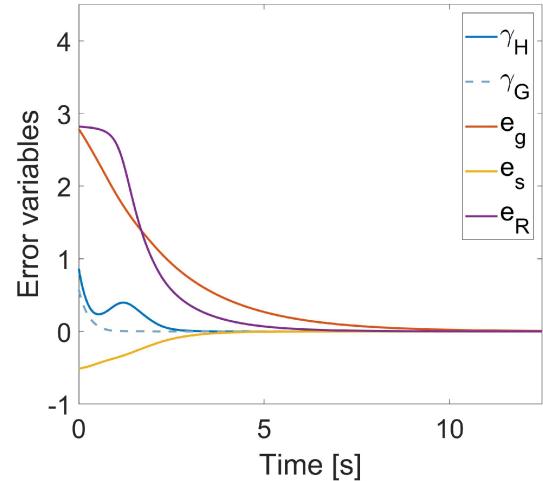


Figura 6.4: Comparación de la configuración de la formación en el instante $t = 0.8$ s al aplicar los dos métodos de control considerados, con $k_G = 2$, en el experimento simulado donde el equipo de robots pasa de una configuración deformada a una configuración rectangular y rotada en un espacio 3D.

La Figura 6.5 permite comparar la evolución de las variables de error. Debido a la distinta formulación de los dos controladores y puesto que en este ejemplo la tarea implica una rotación, el valor inicial de algunos errores varía de un caso a otro. Además, se observa que el error de rotación converge a cero en el mismo tiempo en ambos casos, aunque lo hace de manera más progresiva en el segundo de ellos. Por otro lado, mientras que con las rotaciones infinitesimales el error de conservación de forma se reduce progresivamente, al utilizar el algoritmo de Kabsch se observa una oscilación en los primeros momentos de la simulación, coincidente con el inicio de la rápida disminución del error e_R . Aun así, es importante destacar que, aunque dicho error aumente momentáneamente, nunca alcanzará valores superiores a los de $t = 0$ s. Si esta oscilación se considerase problemática en un experimento concreto, se puede reducir o, incluso, eliminar reduciendo el valor de la ganancia k_{Hd} , aunque esto hará que el cambio de escala y la rotación sean más lentos.



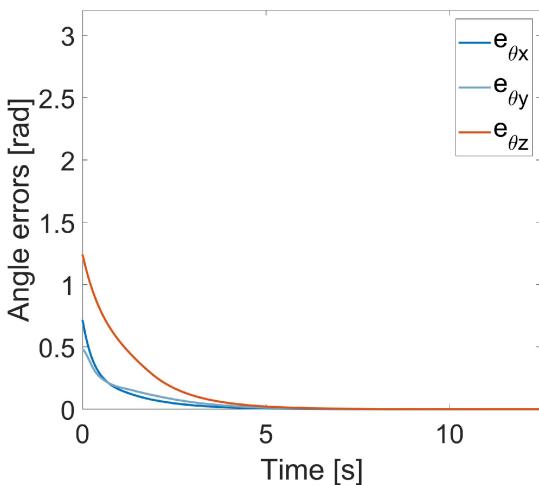
(a) Control con rotaciones infinitesimales



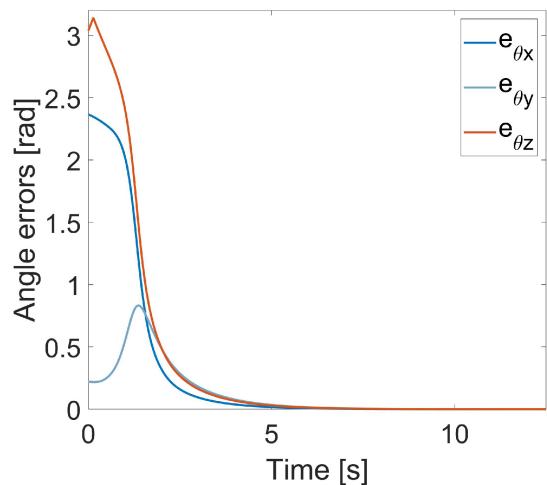
(b) Control con algoritmo de Kabsch

Figura 6.5: Comparación de la evolución de las variables de error al aplicar los dos métodos de control considerados, con $k_G = 2$, en el experimento simulado donde el equipo de robots pasa de una configuración deformada a una configuración rectangular y rotada en un espacio 3D.

Cabe destacar que, a diferencia de los experimentos 2D donde la orientación se describía mediante un único ángulo, en 3D se define la rotación mediante los tres ángulos de Euler. La Figura 6.6 refleja la evolución del error de orientación de estos ángulos en torno a los ejes locales del objeto. Se observa una reducción suave en el caso de las rotaciones infinitesimales frente a una disminución más progresiva en los primeros segundos al usar el algoritmo de Kabsch. Además, en este último caso, el error de orientación en el eje x presenta una oscilación, la cual ocurre en el mismo instante temporal que la oscilación del error de conservación de forma comentado anteriormente.



(a) Control con rotaciones infinitesimales



(b) Control con algoritmo de Kabsch

Figura 6.6: Comparación de la evolución del error de los ángulos de rotación al aplicar los dos métodos de control considerados, con $k_G = 2$, en el experimento simulado donde el equipo de robots pasa de una configuración deformada a una configuración rectangular y rotada en un espacio 3D.

Por último, la Figura 6.7 refleja las trayectorias seguidas por cada robot. Se observa que el controlador que modela las rotaciones como infinitesimales funciona correctamente aunque las rotaciones reales a realizar sean considerables, como en este ejemplo. El uso conjunto de todos los términos de control propuestos permite obtener una evolución adecuada de la formación. Para más resultados, acudir al Anexo D.2.

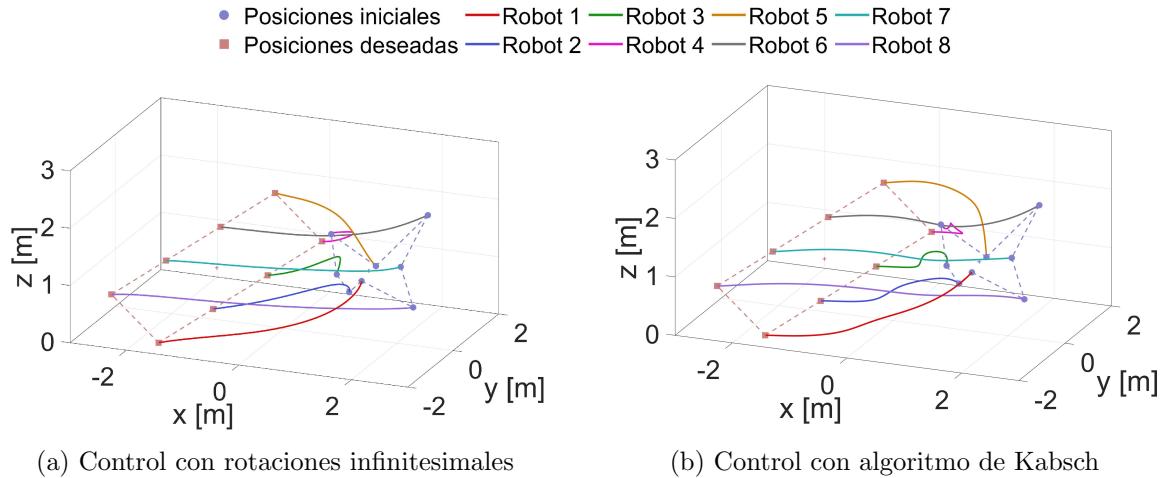


Figura 6.7: Comparación de las trayectorias seguidas por los robots al aplicar los dos métodos de control considerados, con $k_G = 2$, en el experimento simulado donde el equipo de robots pasa de una configuración deformada a una configuración rectangular y rotada en un espacio 3D.

6.3. Transporte de objeto deformable volumétrico

La variedad de objetos deformables es enorme, desde una lámina flexible o una sábana como se ha mostrado en los experimentos realizados anteriormente, hasta un colchón o una almohada. Para verificar que el controlador también es válido en el transporte de objetos deformables volumétricos, se propone el siguiente experimento. En él, un equipo de 6 robots transporta un colchón a lo largo de un pasillo curvo, el cual obliga a la formación a rotar para evitar colisiones y deformaciones.

Para simular este objeto se ha usado el método *Meshless Shape Matching* [16], el cual es apropiado para representar objetos volumétricos. El objeto se ha simulado con las dimensiones típicas de un colchón real ($1.5 \times 2.1 \times 0.3$ metros). Igual que en el ejemplo del Apartado 4.4, se definen manualmente tres *waypoints* intermedios que guían a la formación a través del escenario planteado.

La idea, además de validar el controlador, es demostrar que también se puede aplicar en tareas de transporte que requieren mantener la forma y escala del objeto en reposo.

Por ello, se escogen los siguientes valores para las ganancias: $k_H = 50$, $k_G = 5$, $k_s = 50$, $k_c = 1$ y $k_{Hd} = 1.5$. Puesto que en los apartados anteriores ya se han analizado las diferencias entre el control con rotaciones infinitesimales y con el algoritmo de Kabsch, a continuación solo se van a mostrar los resultados obtenidos con el segundo método.

En la Figura 6.8 puede verse el colchón en un momento concreto del transporte, así como las trayectorias seguidas por cada robot desde su posición inicial hasta la final. El enlace de acceso al vídeo del experimento se encuentra en el Anexo F.7.

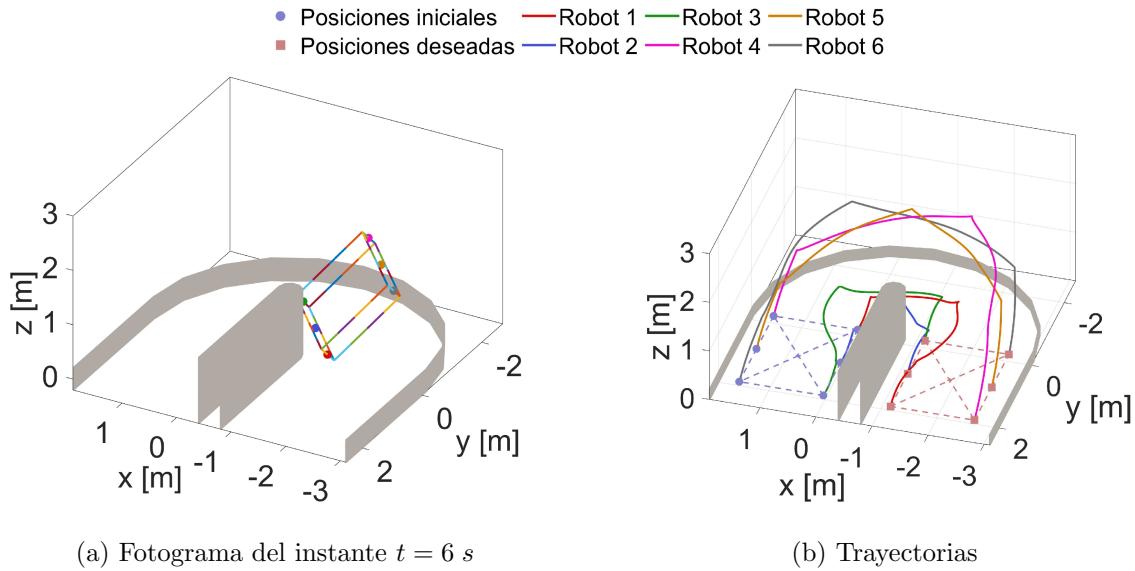
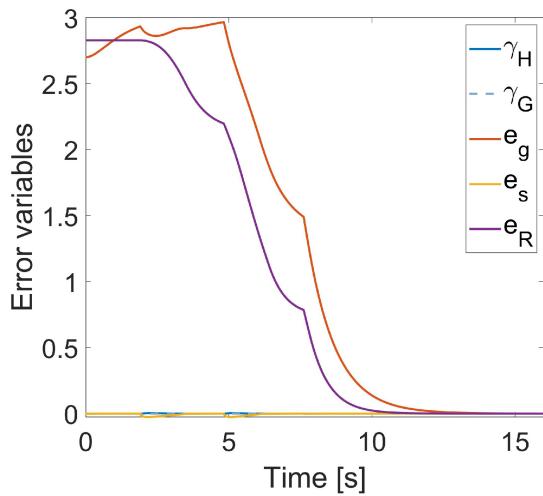


Figura 6.8: Representación de la configuración de la formación en el instante $t = 6 s$ y las trayectorias seguidas por los robots en el experimento simulado donde el equipo de robots transporta un colchón a lo largo de un pasillo curvo en un espacio 3D.

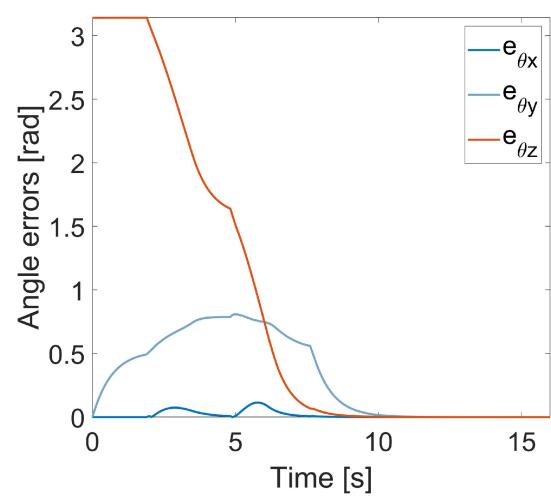
Al estar rotando, el objeto se deformaría de manera descontrolada de no ser por el controlador implementado. En cambio, puede comprobarse que el colchón mantiene su forma y escala original. Esto también se refleja en las gráficas de evolución de los errores de la Figura 6.9.

Se puede observar que los errores de forma y escala son prácticamente nulos durante toda la tarea. En la representación de esta última variable de error se aprecian ligeras oscilaciones en dos momentos concretos de la simulación, coincidentes con los cambios de orientación que realiza el equipo de robots. Tratándose de un colchón, aun siendo un objeto bastante rígido, posee cierta deformabilidad, por lo que estas pequeñas variaciones de escala no supondrían un problema.

En el Anexo D.3 puede verse el movimiento completo del objeto a lo largo del pasillo, así como resultados adicionales.



(a) Evolución de las variables de error



(b) Errores de los tres ángulos de rotación

Figura 6.9: Representación de la evolución de las variables de error y de los ángulos de rotación en el experimento simulado donde el equipo de robots transporta un colchón a lo largo de un pasillo curvo en un espacio 3D.

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones alcanzadas

Al comienzo de esta investigación se estableció como objetivo principal la implementación del controlador de forma con modos de deformación en sistemas multi-robot y su posterior análisis en diferentes entornos.

La primera parte recoge el desarrollo teórico del control aplicado en escenarios 2D, así como los resultados de los diferentes experimentos realizados para su validación. Estos experimentos abarcan desde simples desplazamientos, donde el equipo de robots debe deformarse para alcanzar la configuración final deseada, hasta escenarios más complejos en los que el movimiento de los robots está restringido por obstáculos. Se han podido observar las ventajas que ofrece este controlador frente a otros que no limiten el rango de deformaciones. De esta forma, se garantiza que el objeto no sufra daños que afecten a su integridad física. Además, el resto de parámetros de la configuración también se controlan satisfactoriamente. Como contribución adicional, el trabajo realizado en esta primera parte se ha presentado en el *4th Workshop on Representing and Manipulating Deformable Objects* del congreso internacional *IEEE International Conference on Robotics and Automation* (ICRA) de 2024.

En la segunda parte se explica el proceso de adaptación del controlador a entornos 3D. Al existir mayor libertad de movimiento en estos espacios, el control de rotación es más complejo, por lo que se han planteado dos métodos distintos para ello, como son la aproximación mediante rotaciones infinitesimales y el algoritmo de Kabsch. Ambas técnicas son una buena solución, aunque poseen ciertas diferencias. Por un lado, las

rotaciones infinitesimales implican el uso de aproximaciones de las rotaciones óptimas reales, pero permiten tener un controlador formulado analíticamente. Por otro lado, el algoritmo de Kabsch permite usar las rotaciones óptimas exactas, pero dificulta su tratamiento por ser un método numérico. Se han realizado diversos ensayos en simulación que han posibilitado la comparación de ambos métodos, así como comprobar que el controlador funciona correctamente también en espacios tridimensionales para transportar objetos deformables superficiales y volumétricos.

El uso de Robotarium ha sido esencial para probar y ajustar experimentalmente el controlador propuesto. Además, ha permitido demostrar su viabilidad y efectividad, tanto en simulación como con robots reales. Por otro lado, el hecho de simular el objeto transportado ha facilitado la interpretación de las deformaciones que podría sufrir un objeto real.

Todo el trabajo recogido entre estas páginas permite afirmar que los objetivos marcados se han alcanzado favorablemente. La implementación del controlador en entornos 2D y 3D ha resultado exitosa. Puede considerarse como una buena alternativa de control multi-robot para el transporte de objetos deformables, principalmente de aquellos en los que, por sus propiedades físicas, sea primordial controlar cómo y cuánto pueden deformarse.

7.2. Trabajo futuro

Aunque este TFM engloba muchos elementos relacionados con la robótica móvil y el transporte de objetos deformables, algunos de ellos podrían tratarse más en profundidad en futuras líneas de investigación.

En los experimentos realizados en entornos complejos, los obstáculos eran fijos y los robots podían corregir su trayectoria y esquivarlos gracias a las CBF incorporadas en Robotarium. Para implementar el controlador propuesto en otros entornos más realistas, con mayor cantidad de obstáculos o, incluso, con la presencia de obstáculos móviles, se podría trabajar en la idea de sensorizar los robots. De esta forma, podrían analizar su área de movimiento en cada instante y recalcular sus trayectorias antes de sufrir colisiones, sin necesidad de que el usuario deba predefinir el espacio de trabajo.

Otra posible dirección de trabajo sería la planificación de trayectorias. Aquí, se han usado puntos intermedios, o *waypoints*, para indicar al equipo de robots el camino a seguir. Otra alternativa podría ser el desarrollo de algoritmos que calculen la ruta óptima entre diferentes puntos, teniendo en cuenta la ubicación de los obstáculos, o implementar algoritmos de planificación ya existentes.

Por último, también se podrían considerar en el control las propiedades físicas del objeto transportado, de manera que la formación robótica no se deforme más de lo que el límite elástico permite.

Bibliografía

- [1] F. Rubio, F. Valero y C. Llopis-Albert, «A review of mobile robots: Concepts, methods, theoretical framework, and applications,» *International Journal of Advanced Robotic Systems*, vol. 16, n.º 2, 2019, Article 1729881419839596. DOI: 10.1177/1729881419839596.
- [2] L. E. Parker, D. Rus y G. S. Sukhatme, «Multiple Mobile Robot Systems,» en *Springer Handbook of Robotics*, B. Siciliano y O. Khatib, eds. Cham: Springer International Publishing, 2016, págs. 1335-1384, ISBN: 978-3-319-32552-1. DOI: 10.1007/978-3-319-32552-1_53.
- [3] R. Marcos Saavedra, G. López Nicolás y M. Aranda Calleja, «Control de forma multi-robot en el Robotarium,» TFG, Universidad de Zaragoza, nov. de 2022. dirección: <https://deposita.unizar.es/TAZ/EINA/2022/72072/TAZ-TFG-2022-4642.pdf>.
- [4] S. Wilson, P. Glotfelter, L. Wang et al., «The Robotarium: Globally Impactful Opportunities, Challenges, and Lessons Learned in Remote-Access, Distributed Control of Multirobot Systems,» *IEEE Control Systems Magazine*, vol. 40, n.º 1, págs. 26-44, 2020. DOI: 10.1109/MCS.2019.2949973.
- [5] R. Marcos-Saavedra, M. Aranda y G. López-Nicolás, «Multirobot transport of deformable objects using deformation modes,» *4th Workshop on Representing and Manipulating Deformable Objects, 2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. dirección: https://deformable-workshop.github.io/icra2024/spotlight/02_03_wdo_marcossaavedra_multirobot.pdf (visitado 13-06-2024).
- [6] R. Marcos-Saavedra, M. Aranda y G. López-Nicolás, «RMDO 2024: Multirobot transport of deformable objects using deformation modes,» *4th Workshop on Representing and Manipulating Deformable Objects, 2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. dirección: <https://www.youtube.com/watch?v=Lm27TvNG9Cw> (visitado 13-06-2024).
- [7] R. Herguedas, G. López-Nicolás, R. Aragüés y C. Sagüés, «Survey on multi-robot manipulation of deformable objects,» en *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, págs. 977-984. DOI: 10.1109/ETFA.2019.8868987.

- [8] J. Alonso-Mora, R. Knepper, R. Siegwart y D. Rus, «Local motion planning for collaborative multi-robot manipulation of deformable objects,» en *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, págs. 5495-5502. DOI: 10.1109/ICRA.2015.7139967.
- [9] D. Koung, O. Kermorgant, I. Fantoni y L. Belouaer, «Cooperative Multi-Robot Object Transportation System Based on Hierarchical Quadratic Programming,» *IEEE Robotics and Automation Letters*, vol. 6, n.º 4, págs. 6466-6472, 2021. DOI: 10.1109/LRA.2021.3092305.
- [10] R. Herguedas, G. López-Nicolás y C. Sagüés, «Multirobot Transport of Deformable Objects With Collision Avoidance,» *IEEE Systems Journal*, vol. 17, n.º 2, págs. 3224-3234, 2023. DOI: 10.1109/JSYST.2022.3213972.
- [11] J. Hu, W. Liu, H. Zhang, J. Yi y Z. Xiong, «Multi-Robot Object Transport Motion Planning With a Deformable Sheet,» *IEEE Robotics and Automation Letters*, vol. 7, n.º 4, págs. 9350-9357, 2022. DOI: 10.1109/LRA.2022.3191190.
- [12] C. Yang, G. Sue, Z. Li et al., «Collaborative Navigation and Manipulation of a Cable-Towed Load by Multiple Quadrupedal Robots,» *IEEE Robotics and Automation Letters*, vol. 7, n.º 4, págs. 10 041-10 048, oct. de 2022. DOI: 10.1109/LRA.2022.3191170.
- [13] M. Aranda, R. Aragüés y G. López-Nicolás, «Combined Leaderless Control of Translational, Shape-Preserving, and Affine Multirobot Formations,» *IEEE Robotics and Automation Letters*, vol. 8, págs. 7567-7574, sep. de 2023. DOI: 10.1109/LRA.2023.3316889.
- [14] R. Herguedas, M. Aranda, G. López-Nicolás, C. Sagüés e Y. Mezouar, «Double-Integrator Multirobot Control With Uncoupled Dynamics for Transport of Deformable Objects,» *IEEE Robotics and Automation Letters*, vol. 8, n.º 11, págs. 7623-7630, 2023. DOI: 10.1109/LRA.2023.3320430.
- [15] A. Pentland y J. Williams, «Good vibrations: modal dynamics for graphics and animation,» *SIGGRAPH Comput. Graph.*, vol. 23, n.º 3, págs. 207-214, jul. de 1989, ISSN: 0097-8930. DOI: 10.1145/74334.74355.
- [16] M. Müller, B. Heidelberger, M. Teschner y M. Gross, «Meshless deformations based on shape matching,» *ACM Trans. Graph.*, vol. 24, n.º 3, págs. 471-478, jul. de 2005, ISSN: 0730-0301. DOI: 10.1145/1073204.1073216.
- [17] P. Güler, A. Pieropan, M. Ishikawa y D. Kragic, «Estimating deformability of objects using meshless shape matching,» en *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, págs. 5941-5948. DOI: 10.1109/IROS.2017.8206489.

- [18] I. Cuiral-Zueco y G. López-Nicolás, «Multiscale Procrustes-Based 3-D Shape Control,» *IEEE/ASME Transactions on Mechatronics*, vol. 29, n.º 3, págs. 1738-1748, 2024. DOI: 10.1109/TMECH.2023.3325934.
- [19] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl y H.-P. Seidel, «Laplacian surface editing,» en *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, Association for Computing Machinery, 2004, págs. 175-184, ISBN: 3905673134. DOI: 10.1145/1057432.1057456.
- [20] O. Sorkine-Hornung y M. Rabinovich, «Least-squares rigid motion using SVD,» *ETH Department of Computer Science*, 2017.
- [21] M. Aranda, J. Sanchez, J. A. C. Ramon e Y. Mezouar, «Robotic Motion Coordination Based on a Geometric Deformation Measure,» *IEEE Systems Journal*, vol. 16, n.º 3, págs. 3689-3699, 2022. DOI: 10.1109/JSYST.2021.3107779.
- [22] A. Valdeolmillos, C. Sagüés y R. Herguedas, «Manipulation of Deformable Objects with a Multi-robot System,» en *Robot 2023: Sixth Iberian Robotics Conference*, L. Marques, C. Santos, J. L. Lima, D. Tardioli y M. Ferre, eds., Springer Nature Switzerland, 2024, págs. 375-386, ISBN: 978-3-031-58676-7.
- [23] O. Sorkine y M. Alexa, «As-Rigid-As-Possible Surface Modeling,» en *Geometry Processing*, A. Belyaev y M. Garland, eds., The Eurographics Association, 2007, ISBN: 978-3-905673-46-3. DOI: 10.2312/SGP/SGP07/109-116.

Anexos

Anexo A

Workshop 2024

Todo el trabajo realizado durante el desarrollo, implementación y validación del controlador propuesto en entornos bidimensionales ha servido de base para la redacción del artículo de investigación recogido en las siguientes páginas. Este *paper*, también accesible en [5], se ha presentado en el *4th Workshop on Representing and Manipulating Deformable Objects* del *IEEE International Conference on Robotics and Automation* (ICRA) de 2024.

Además, se ha grabado un vídeo explicativo del artículo, presentado en el propio *workshop* y publicado en su canal de YouTube [6].

Multirobot transport of deformable objects using deformation modes

Raquel Marcos-Saavedra, Miguel Aranda, and Gonzalo López-Nicolás

Abstract— We present a formation controller for transporting deformable objects in 2D space with a team of mobile robots. We assume the deformation of the transported object is determined by the deformation of the robotic formation. The goal is to reach a target configuration consisting of a desired shape, scale, position and orientation, allowing linear and quadratic deformations of the robotic formation relative to the target configuration. The use of this range of deformation modes enables preserving the integrity of the object while making the transport system highly flexible. The controller is tested in simulations and in real experiments with unicycle robots.

I. INTRODUCTION

Transporting a deformable object with a team of robots can be a challenging task that requires high coordination, especially if the object is large, heavy or fragile, or if the trajectory to follow requires the team of robots to execute very specific actions. Formation control allows multirobot systems to manipulate these objects with very accurate movements to prevent damage during the transport.

The manipulation of deformable objects with multiple robots is a field widely covered in prior works [1]. In this context, transport tasks have been considered in different scenarios [2]–[6]. Some related work [7] presents the coordinated motion of the team of robots, modelled by single-integrator dynamics, through a linear combination of translation, shape-preserving transformation and affine transformation of a reference configuration. This combination enables the robots to carry out efficient rotation and resizing maneuvers. Other studies propose a solution exploiting measures of deformation with a linear control law considering single-integrator dynamics [8] or assuming double-integrator dynamics [9] to add inertial effects to the system. The controller in [9] allows driving a deformable object to a desired state by driving the team to a target configuration, defined as a combination of shape, scale, position and orientation in 2D space.

Deformation modes have been used to model deformable objects in computer graphics. Two examples are [10], which exploited the Finite Element Method, and [11], which proposed a geometric approach based on shape matching. This

The authors are with Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain. E-mail: 739917@unizar.es; miguel.aranda@unizar.es; gonlopez@unizar.es.

This work was supported via project REMAIN - S1/1.1/E0111 (Interreg Sudoe Programme, ERDF), and via projects PID2021-124137OB-I00 and TED2021-130224B-I00 funded by MCIN/AEI/10.13039/501100011033, by ERDF A way of making Europe and by the European Union NextGenerationEU/PRTR. R. Marcos-Saavedra was supported by Cátedra de Transformación Industrial (Universidad de Zaragoza - Gobierno de Aragón). M. Aranda was supported via a María Zambrano Fellowship funded by the Spanish Ministry of Universities and by the European Union-NextGenerationEU.

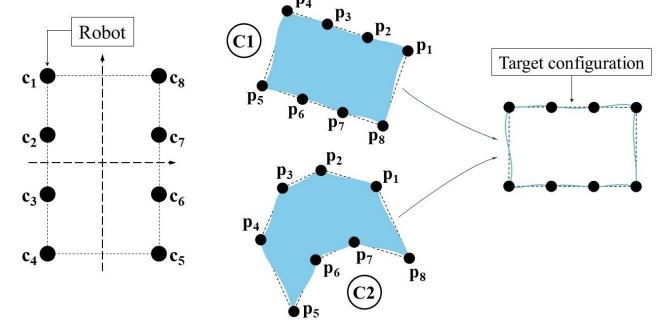


Fig. 1. (Left) Reference configuration of the formation. (Right) Representation of two achievable configurations (C1 and C2) during the transport of a deformable object by a team of mobile robots steered with the proposed controller to a target configuration, defined as a combination of desired shape, scale, position and orientation. If the task does not require that the object's shape is modified, the formation will be kept in a shape-preserving configuration (C1). However, if the formation needs to deform the object, the deformed formation will be constrained to linear and quadratic deformation modes allowing stretching, shearing, bending and twisting while avoiding other unsuitable and unpredictable deformation patterns (C2).

latter approach was used in [12] to estimate deformability in robotics applications. Modal analysis has also been recently exploited for shape control [13]. Linear deformation modes (i.e., stretching and shearing) were considered in [9] to control deformation during the transport, enhancing flexibility.

In this paper, we propose a method to transport a deformable object grasped around its contour by multiple mobile robots that allows the agents to perform maneuvers that deform the object only with linear and quadratic deformation modes during the trajectory. Some situations require the transported object to be deformed linearly, stretching or shearing it, as considered in [9]. Others, however, also need to deform it in a quadratic way, especially when the task entails changes in direction, causing the object to bend or twist. Therefore, we introduce quadratic deformation modes, inspired by [11], to increase adaptability relative to [9].

II. MULTIROBOT CONTROL WITH DEFORMATION MODES

We consider a formation of N robots moving in a 2D space grasping the transported object through rotational joints, as illustrated in Fig. 1. We denote the position of robot $i \in \{1, \dots, N\}$ by \mathbf{p}_i . We assume single-integrator dynamics, i.e., $\dot{\mathbf{p}}_i = \mathbf{u}_i$, with \mathbf{u}_i the control input. We group for the full team $\mathbf{p} = [\mathbf{p}_1^\top, \dots, \mathbf{p}_N^\top]^\top \in \mathbb{R}^{2N}$ and $\mathbf{u} = [\mathbf{u}_1^\top, \dots, \mathbf{u}_N^\top]^\top \in \mathbb{R}^{2N}$. The control strategy is based on separately controlling each configuration parameter (shape, scale, position and orientation) and using a linear combination of those controllers.

As in [9], we assume that by suitably controlling the shape and scale of the team of robots we can control the deformation of the object and maintain its integrity. This assumption is valid, e.g., for highly deformable objects whose shape adapts to the shape of the team of robots.

Shape Control. We define $\mathbf{c}_i = [c_{ix}, c_{iy}]^\top$ as the position of the robot i in the reference configuration. Note that the target configuration is defined with the same shape as the reference one up to a scale, translation and rotation (Fig. 1). In order to control the shape of the formation during the transport, two types of configurations are considered. The shape-preserving configuration keeps the team in the same shape as the reference configuration. Let us define the following matrix:

$$\mathbf{C}_H = \mathbf{K} \begin{bmatrix} c_{1x} & c_{1y} & \dots & c_{Nx} & c_{Ny} \\ -c_{1y} & c_{1x} & \dots & -c_{Ny} & c_{Nx} \end{bmatrix}^\top \in \mathbb{R}^{2N \times 2}, \quad (1)$$

where $\mathbf{K} = (\mathbf{I}_N - (1/N)\mathbf{1}_N\mathbf{1}_N^\top) \otimes \mathbf{I}_2 \in \mathbb{R}^{2N \times 2N}$ is a centering matrix which translates the centroid to zero, \mathbf{I}_N is the $N \times N$ identity matrix, $\mathbf{1}_N$ is a column vector of N ones and \otimes denotes the Kronecker product.

Notice that for any $\mathbf{h}_H = [h_{H1}, h_{H2}]^\top$ the condition $\mathbf{Kp} - \mathbf{C}_H \mathbf{h}_H = \mathbf{0}$ is equivalent to $\mathbf{p}_i = [[h_{H1}, h_{H2}]^\top, [-h_{H2}, h_{H1}]^\top] \mathbf{c}_i$ for every robot i with the sets of points \mathbf{p}_i and \mathbf{c}_i both having zero centroid. This represents a rotation and uniform scaling of the reference configuration [9]. Note that we use centering for optimality [9], [11]. Therefore, if this condition is satisfied while the robots are moving, the team's shape will be kept. To define \mathbf{h}_H we propose to use a least-squares shape alignment strategy: i.e., we choose \mathbf{h}_H so that $\|\mathbf{Kp} - \mathbf{C}_H \mathbf{h}_H\|$ is minimum, being $\|\cdot\|$ the Euclidean norm. We define henceforth $\mathbf{h}_H = \mathbf{C}_H^+ \mathbf{Kp} = \mathbf{C}_H^+ \mathbf{p}$, as $\mathbf{C}_H^+ \mathbf{K} = \mathbf{C}_H^+$, where $+$ denotes the Moore-Penrose inverse. We can formulate the following cost function associated with shape preservation:

$$\gamma_H = \frac{1}{2} \|\mathbf{Kp} - \mathbf{C}_H \mathbf{h}_H\|^2 = \frac{1}{2} \mathbf{p}^\top \mathbf{A}_H \mathbf{p}, \quad (2)$$

where $\mathbf{A}_H = \mathbf{K} - \mathbf{C}_H \mathbf{C}_H^+$. Note that \mathbf{A}_H is constant, symmetric, idempotent and positive semidefinite. Then, we propose a controller for preserving the shape of the team following the negative gradient of γ_H :

$$\mathbf{u}_H = -k_H \mathbf{A}_H \mathbf{p}, \quad (3)$$

where k_H is a positive control gain.

The second configuration we consider is the deformed configuration which is expressed by deformation modes up to order two, allowing the formation to deform in a controlled way. Inspired by [11], we define this deformation with linear (\mathbf{L}_i), quadratic (\mathbf{Q}_i) and mixed (\mathbf{M}_i) terms. Analogously to \mathbf{C}_H above, we can define the matrix

$$\mathbf{C}_G = \mathbf{K} \begin{bmatrix} \mathbf{L}_1 & \mathbf{Q}_1 & \mathbf{M}_1 \\ \vdots & \vdots & \vdots \\ \mathbf{L}_N & \mathbf{Q}_N & \mathbf{M}_N \end{bmatrix} \in \mathbb{R}^{2N \times 10}, \quad (4)$$

$$\mathbf{L}_i = \begin{bmatrix} c_{ix} & c_{iy} & 0 & 0 \\ 0 & 0 & c_{ix} & c_{iy} \end{bmatrix} \in \mathbb{R}^{2 \times 4}, \quad (5)$$

$$\mathbf{Q}_i = \begin{bmatrix} c_{ix}^2 & c_{iy}^2 & 0 & 0 \\ 0 & 0 & c_{ix}^2 & c_{iy}^2 \end{bmatrix} \in \mathbb{R}^{2 \times 4}, \quad (6)$$

$$\mathbf{M}_i = \begin{bmatrix} c_{ix} c_{iy} & 0 \\ 0 & c_{ix} c_{iy} \end{bmatrix} \in \mathbb{R}^{2 \times 2}, \quad (7)$$

where \mathbf{L}_i terms can only represent shear and stretch, whereas \mathbf{Q}_i and \mathbf{M}_i terms can represent bend and twist.

Similarly to the shape-preserving control, we choose a vector $\mathbf{h}_G \in \mathbb{R}^{10}$ which minimizes $\|\mathbf{Kp} - \mathbf{C}_G \mathbf{h}_G\|$. Then, we propose a deformation controller following the negative gradient of a cost function $\gamma_G = \frac{1}{2} \mathbf{p}^\top \mathbf{A}_G \mathbf{p}$:

$$\mathbf{u}_G = -k_G \mathbf{A}_G \mathbf{p}, \quad (8)$$

being k_G a positive control gain and $\mathbf{A}_G = \mathbf{K} - \mathbf{C}_G \mathbf{C}_G^+$, which is constant, symmetric, idempotent and positive semidefinite.

Scale Control. To fully control the deformation we also control the scale of the team using the variable $s = \|\mathbf{h}_H\|$. For achieving the desired scale, s_d , we propose the following control term, where k_s is a positive control gain and $s > 0$ can be assumed [9]:

$$\mathbf{u}_s = k_s (s_d - s) (1/s) \mathbf{C}_H \mathbf{h}_H. \quad (9)$$

Translation and Rotation Control. Translation and rotation of the formation do not alter the relative positions of the robots, so they do not affect the object's deformation. The translation controller is responsible for driving the team of robots as a whole to achieve a desired absolute position, \mathbf{g}_d , of the formation centroid $\mathbf{g} = \frac{1}{N} [\mathbf{p}_1, \dots, \mathbf{p}_N] \mathbf{1}_N$. The rotation controller rotates the shape around the formation centroid until the desired angle, θ_d , is reached. The angle can be obtained as $\theta = \text{atan2}(h_{H2}, h_{H1})$. Control terms to achieve these transformations are, respectively,

$$\mathbf{u}_c = k_c \mathbf{1}_N \otimes (\mathbf{g}_d - \mathbf{g}), \quad (10)$$

$$\mathbf{u}_\theta = k_\theta (\theta_d - \theta) (\mathbf{I}_N \otimes \mathbf{S}) \mathbf{C}_H \mathbf{h}_H, \quad (11)$$

where k_c and k_θ are positive control gains and $\mathbf{S} = [[0, 1]^\top, [-1, 0]^\top]$. In practice it is possible to take $\theta_d = 0$ for convenience and without loss of generality.

Full Formation Controller. The full control law results from the linear combination of the individual controllers:

$$\mathbf{u} = \mathbf{u}_H + \mathbf{u}_G + \mathbf{u}_s + \mathbf{u}_c + \mathbf{u}_\theta. \quad (12)$$

III. EXPERIMENTAL VALIDATION

We validate our controller using the Robotarium [14] with multiple unicycle robots. Note that dynamic model conversion and avoidance of collisions (between agents, and with obstacles) are handled by the Robotarium. We define error variables for position, scale and orientation as $\mathbf{e}_g = \mathbf{g} - \mathbf{g}_d$, $\mathbf{e}_s = s - s_d$, $\mathbf{e}_\theta = \theta - \theta_d$, respectively.

We first conduct simulations for a formation of twelve robots manipulating a deformable sheet. The object is modelled with the As-Rigid-As-Possible (ARAP) technique [15]. The results are illustrated in Fig. 2. The videos of the simulations can be seen in [16].

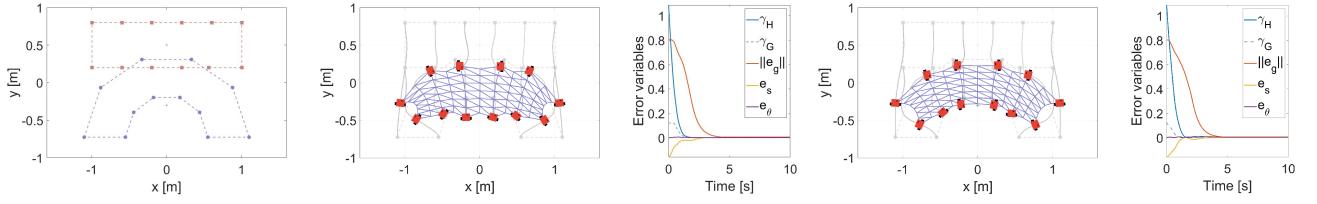


Fig. 2. Simulation results. From left to right, the plots are: 1st, initial configuration (blue circles, bottom) and target configuration (red squares, top); 2nd and 3rd, robot paths and error variables in Case 1; 4th and 5th, robot paths and error variables in Case 2. The configuration of the formation at instant $t = 0.8$ [s] is overlapped on the paths for both cases, showing that only in Case 2 the deformation tends to preserve a quadratic, bending-like pattern. In both cases, control gains are $k_H = 5$, $k_c = 0.5$, $k_s = 2$, and $k_\theta = 0.15$.

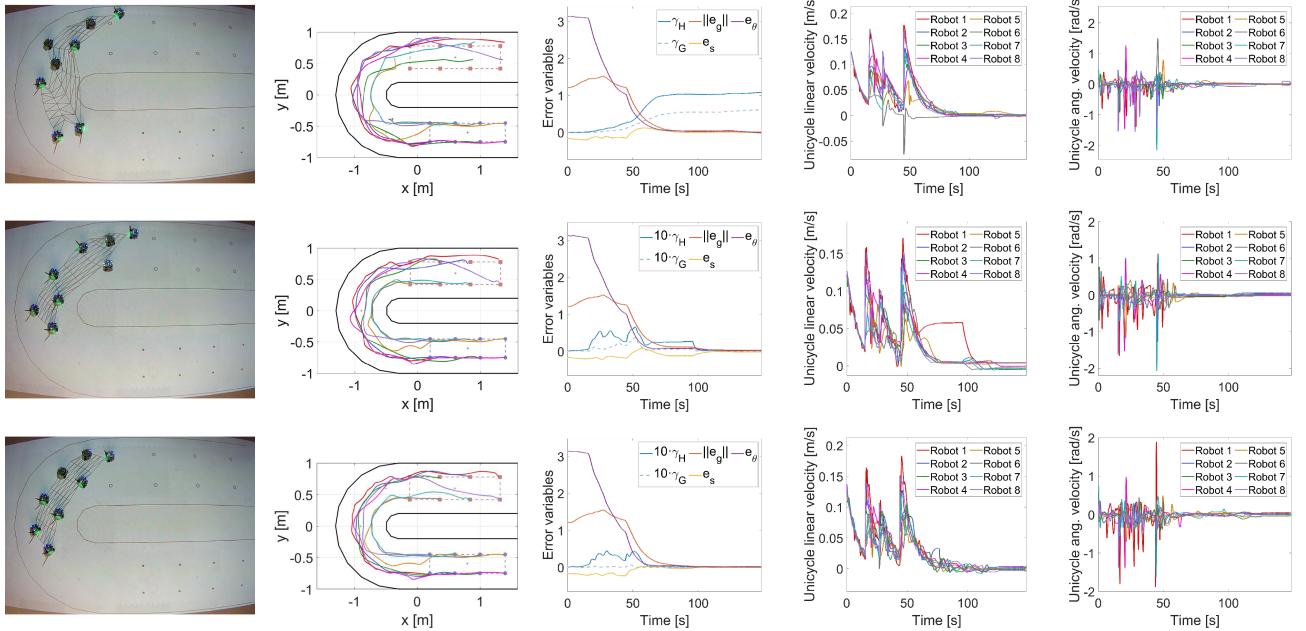


Fig. 3. Experimental results. For each row, the five plots from left to right are: representative top-view snapshots, robot paths, control errors, unicycle linear velocities, and unicycle angular velocities. Each row corresponds to a different shape-control strategy: 1st, the shape is not controlled in any way, $k_H = k_G = 0$, and the shape errors (γ_H , γ_G) do not converge to zero; 2nd, deformation is not controlled, $k_H \neq 0$ and $k_G = 0$; and 3rd, deformation is controlled, $k_H \neq 0$ and $k_G \neq 0$. In the 2nd and 3rd cases, although the shape cannot be preserved, all the errors reach zero values, but only in the 3rd strategy the object deforms in a controlled manner. The improvement of the parameters in the 3rd case with respect to the 2nd one can be seen in the error evolution between 50–100 [s]. The values of the remaining control gains are $k_c = k_s = k_\theta = 0.1$ in the three tests.

We consider a situation where a bent initial configuration has to be driven to a straight target one. We test two cases. For Case 1 we choose $k_G = 0$. The team of robots quickly approaches the same shape as the reference configuration, but deforming the object in an uncontrolled way. For Case 2 we select $k_G = 10$. This allows the formation to maintain a quadratic deformation. The movements are more efficient than in Case 1, producing gradual changes and staying close to a quadratic deformation pattern during the transient period.

Finally, we test the proposed controller in a real scenario with obstacles, where a team of eight robots transports a simulated sheet (also modelled with ARAP) along a curved corridor to a target configuration (Fig. 3). To achieve the experiment's goal we manually define three intermediate waypoints, as a combination of shape, scale, position and orientation, to guide the team during the task. The first

strategy we test is to control the formation scale, position and orientation, but not the shape. Therefore, all errors converge to zero except those related to shape. The second strategy consists in applying $k_H = 0.2$ while maintaining $k_G = 0$, so that the formation tends to preserve the shape of the object but the deformations, which appear inevitably due to the constraints of this particular scenario, are uncontrolled. In the third strategy, we use $k_H = 0.1$ and $k_G = 1$. In this case, the object deforms in a controlled way, preserving its integrity. In the last two strategies all error variables converge to zero. These results support the interest of the proposed approach based on linear and quadratic deformation modes. Video results of these experiments are in [16].

Future work may involve the formal analysis of the controller, or the extension to 3D, with a different formulation of the rotation and uniform scaling transformation.

REFERENCES

- [1] R. Herguedas, G. López-Nicolás, R. Aragüés, and C. Sagüés, “Survey on multi-robot manipulation of deformable objects,” in *Proc. IEEE Int. Conf. Emerg. Technol. Fact. Autom.*, 2019, pp. 977–984.
- [2] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, “Local motion planning for collaborative multi-robot manipulation of deformable objects,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 5495–5502.
- [3] D. Koung, O. Kermorgant, I. Fantoni and L. Belouaer, “Cooperative multi-robot object transportation system based on hierarchical quadratic programming,” in *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6466–6472, 2021.
- [4] R. Herguedas, G. López-Nicolás, and C. Sagüés, “Multirobot transport of deformable objects with collision avoidance,” *IEEE Syst. J.*, vol. 17, no. 2, pp. 3224–3234, 2023.
- [5] J. Hu, W. Liu, H. Zhang, J. Yi, and Z. Xiong, “Multi-robot object transport motion planning with a deformable sheet,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9350–9357, 2022.
- [6] C. Yang, G. N. Sue, Z. Li, L. Yang, H. Shen, Y. Chi, A. Rai, J. Zeng, and K. Sreenath, “Collaborative navigation and manipulation of a cable-towed load by multiple quadrupedal robots,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10041–10048, 2022.
- [7] M. Aranda, R. Aragüés and G. López-Nicolás, “Combined leaderless control of translational, shape-preserving, and affine multirobot formations,” in *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 7567–7574, 2023.
- [8] M. Aranda, J. A. Corrales and Y. Mezouar, “Deformation-based shape control with a multirobot system,” *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 2174–2180.
- [9] R. Herguedas, M. Aranda, G. López-Nicolás, C. Sagüés and Y. Mezouar, “Double-integrator multirobot control with uncoupled dynamics for transport of deformable objects,” in *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 7623–7630, 2023.
- [10] A. Pentland and J. Williams, “Good vibrations: modal dynamics for graphics and animation,” *SIGGRAPH Comput. Graph.*, vol. 23, no. 3, pp. 207–214, 1989.
- [11] M. Müller, B. Heidelberger, M. Teschner and M. Gross, “Meshless deformations based on shape matching,” in *ACM Trans. Graph.*, vol. 24, pp. 471–478, 2005.
- [12] P. Güler, A. Pieropan, M. Ishikawa, and D. Kragic, “Estimating deformability of objects using meshless shape matching,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 5941–5948, 2017.
- [13] B. Yang, B. Lu, W. Chen, F. Zhong, and Y.-H. Liu, “Model-free 3-D shape control of deformable objects using novel features based on modal analysis,” *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 3134–3153, 2023.
- [14] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, “The Robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems,” *IEEE Control Syst. Mag.*, vol. 40, no. 1, pp. 26–44, 2020.
- [15] O. Sorkine and M. Alexa, “As-rigid-as-possible surface modeling,” in *Proc. Eurographics Symp. Geometry Process.*, 2007, pp. 109–116.
- [16] R. Marcos-Saavedra, M. Aranda and G. López-Nicolás, “Multirobot transport of deformable objects using deformation modes,” Accessed: May 6, 2024. [Online]. Available: https://youtube.com/playlist?list=PLjm-KzNA87CoOYVSBtOoxgU3No-d4tw-K&si=d_aZh1NJCu0BdQIr.

Anexo B

Ejemplos de deformaciones alcanzables

Este Anexo recoge una serie de gráficas que representan diferentes ejemplos de deformación que sufre una configuración dada al implementar el controlador de deformaciones, \mathbf{u}_G . El objetivo es reflejar la influencia de cada término de deformación (l_i , q_i y m_i) en el cambio de forma respecto de la configuración de referencia, \mathbf{c} .

Cabe destacar que los resultados mostrados a continuación son específicos de la situación concreta considerada. Cualquier modificación en la configuración de referencia original puede hacer que los términos de deformación actúen de manera diferente.

Dado el vector $\mathbf{h}_G = [l_1 \ l_2 \ l_3 \ l_4 \ q_1 \ q_2 \ q_3 \ q_4 \ m_1 \ m_2]^T$, la configuración deformada se obtiene como $\mathbf{c}_{def} = \mathbf{c} + \mathbf{C}_G \mathbf{h}_G$, con \mathbf{C}_G calculada según el Apartado 3.1.1. Se considera una configuración de referencia rectangular y ligeramente rotada sobre la que se aplican diferentes combinaciones de los términos de deformación. La Figura B.1 muestra esta configuración de referencia sin deformar y la forma que alcanzaría al aplicar un valor unitario a todos los términos de \mathbf{h}_G .

En la Figura B.2 se ilustran las configuraciones obtenidas al deformar la configuración de referencia según los términos lineales, los términos cuadráticos y los términos mixtos. Las gráficas de las Figuras B.3, B.4 y B.5 representan el efecto que tiene cada término individual en la deformación de la configuración de referencia. Se puede comprobar que los términos lineales generan *stretch* y *shear*. Concretamente, los términos l_1 y l_4 transforman la configuración según el primer modo de deformación, mientras que l_2 y l_3 lo hacen según el segundo de ellos. Por otro lado, los términos cuadráticos producen *bend* y los mixtos *twist*.

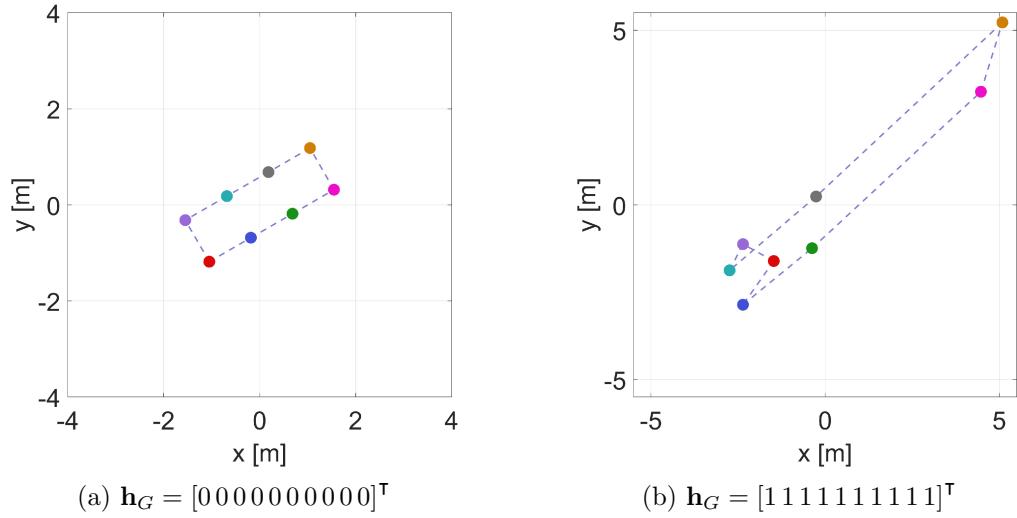


Figura B.1: Representación de la configuración de referencia (izquierda) y la configuración de referencia deformada, \mathbf{c}_{def} , al aplicar todos los términos de deformación (derecha).

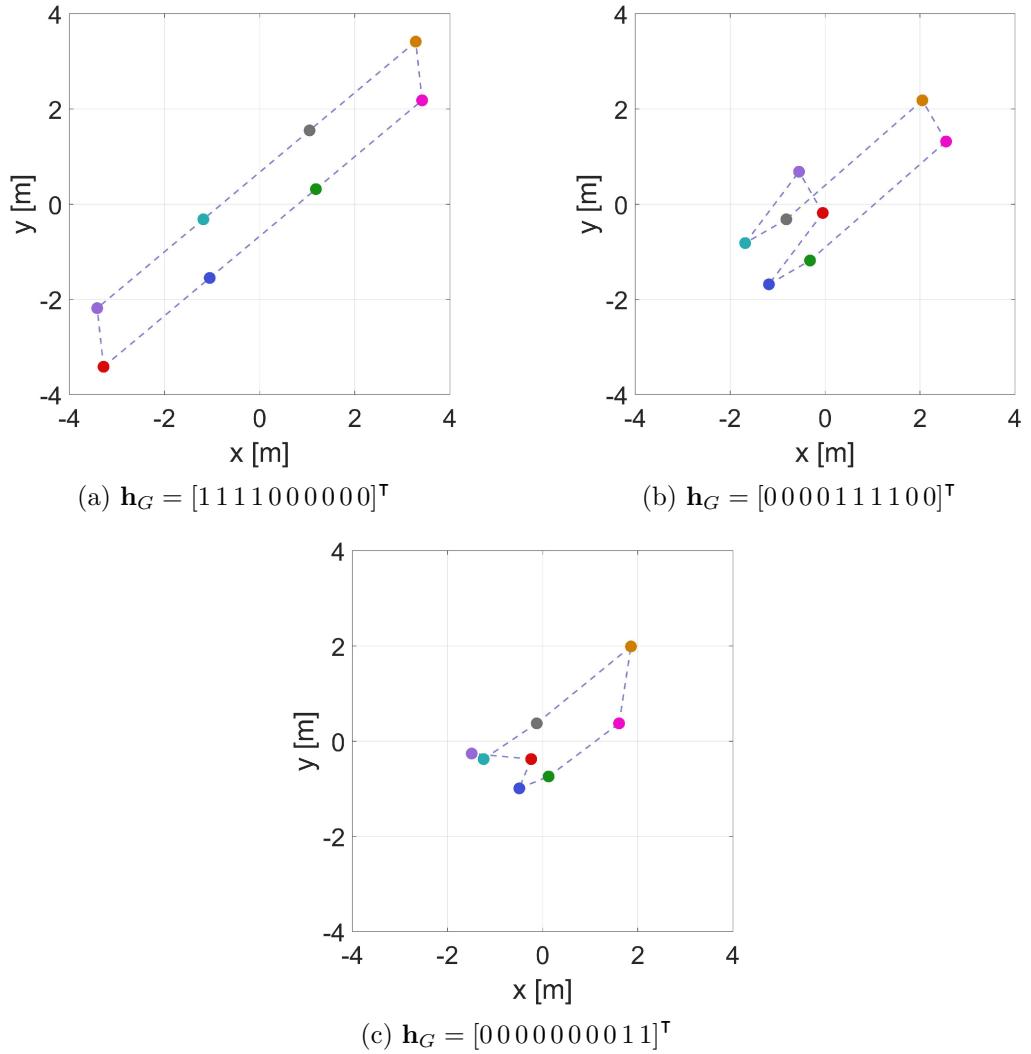


Figura B.2: Representación de la configuración de referencia deformada, \mathbf{c}_{def} , al aplicar los términos lineales (izquierda), cuadráticos (centro) y mixtos (derecha).

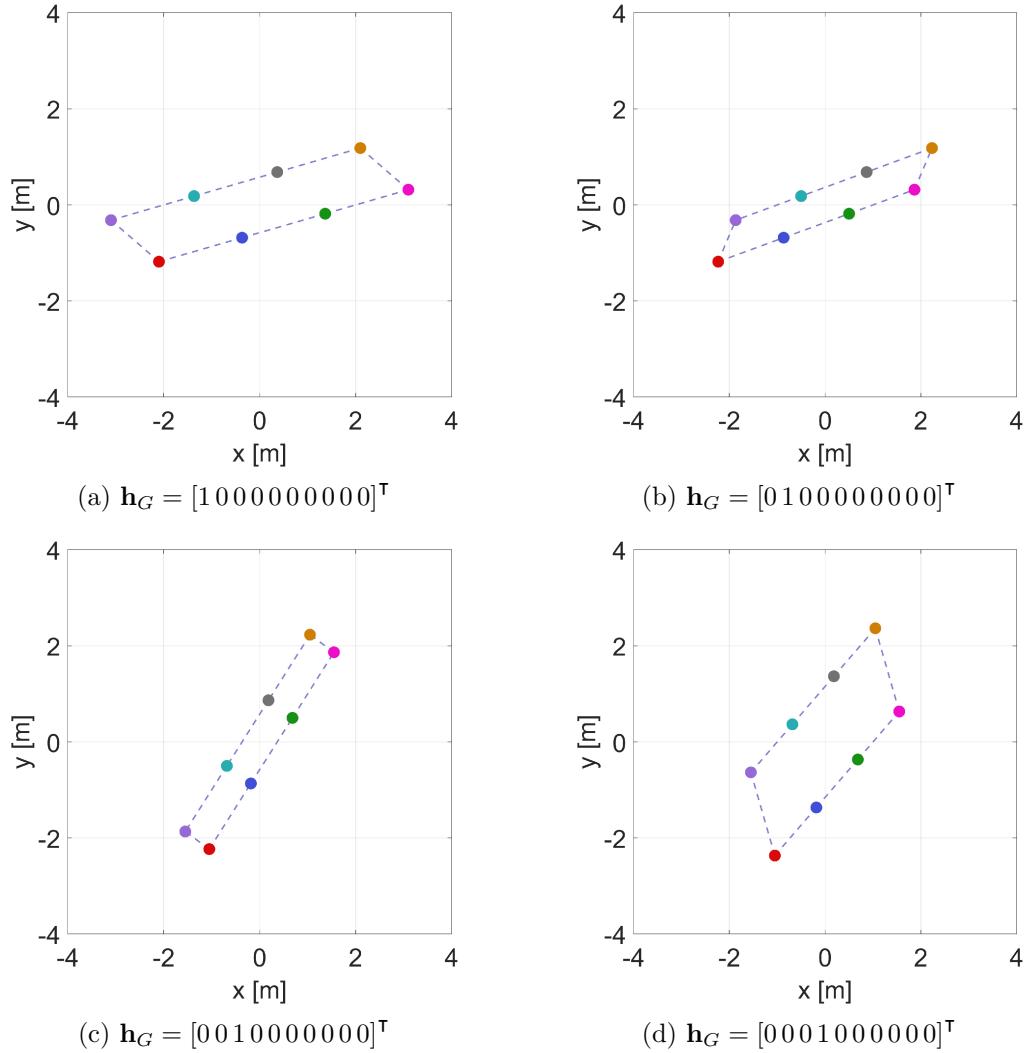


Figura B.3: Representación de la configuración de referencia deformada, \mathbf{c}_{def} , al aplicar cada uno de los términos lineales individualmente. Modos de deformación *stretch* y *shear*.

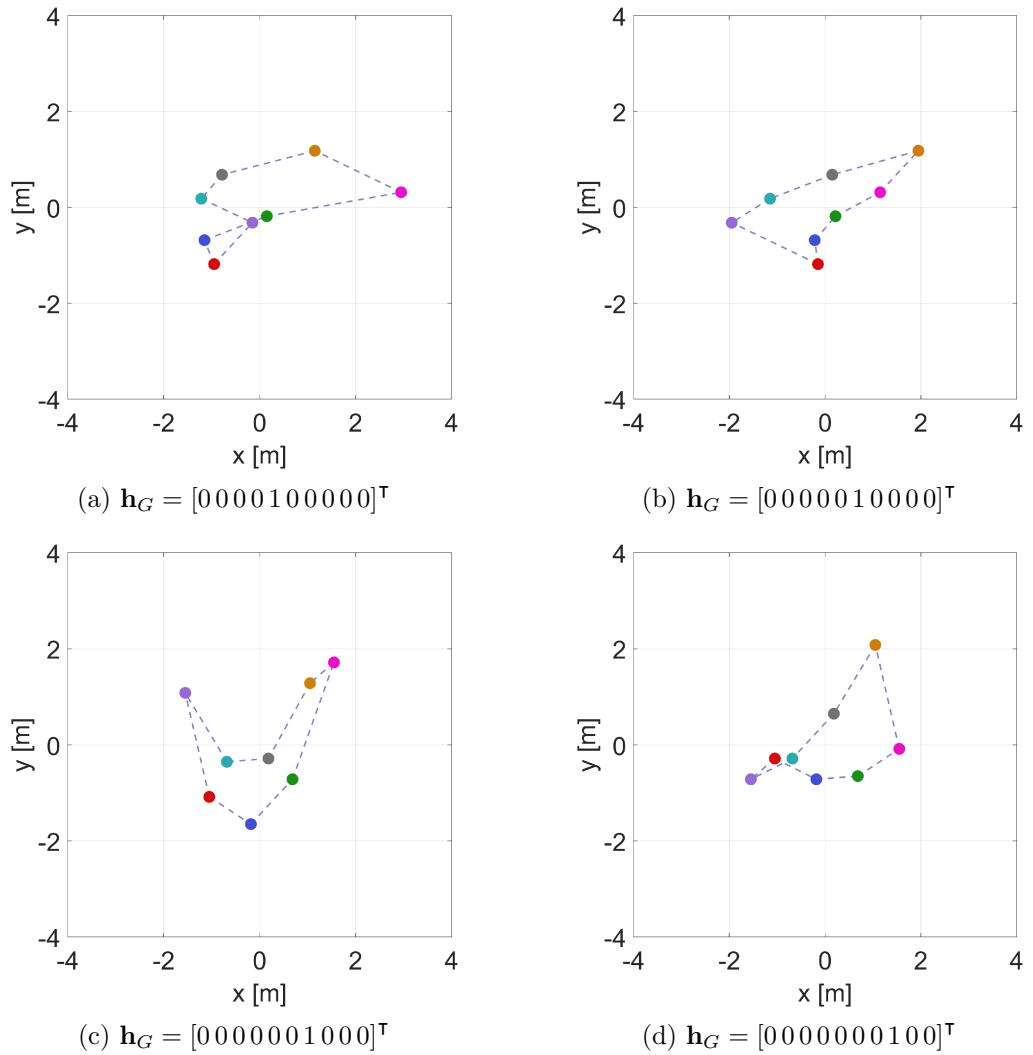


Figura B.4: Representación de la configuración de referencia deformada, \mathbf{c}_{def} , al aplicar cada uno de los términos cuadráticos individualmente. Modo de deformación *bend*.

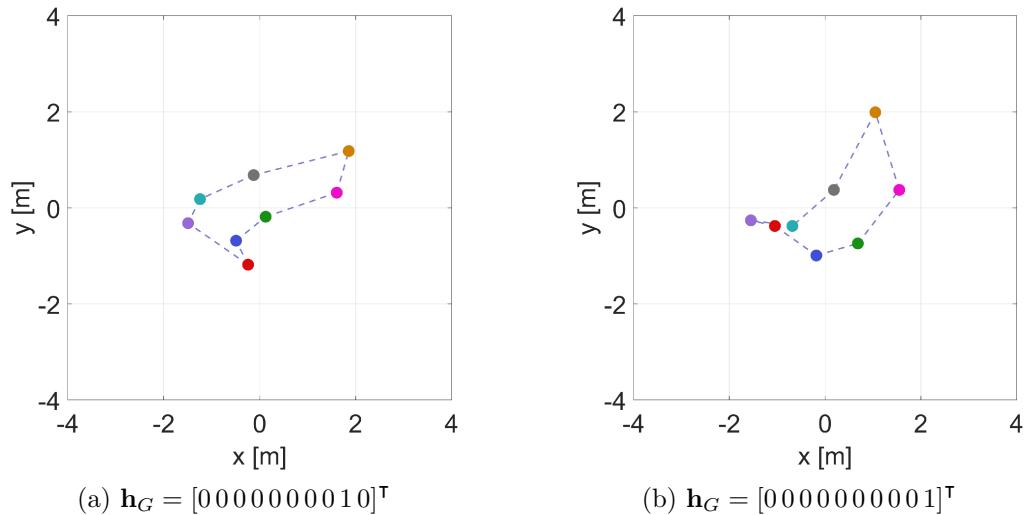


Figura B.5: Representación de la configuración de referencia deformada, \mathbf{c}_{def} , al aplicar cada uno de los términos mixtos individualmente. Modo de deformación *twist*.

Anexo C

Resultados gráficos en entornos 2D

C.1. Gráficas de la simulación 4.2

Se presentan el resto de resultados obtenidos en la simulación del transporte de un objeto deformable en un entorno 2D libre de obstáculos. Las Figuras C.1 y C.2 permiten comparar la deformación sufrida por el objeto durante la tarea en los dos casos considerados, observando que en el Caso 2 esta deformación sigue patrones cuadráticos.

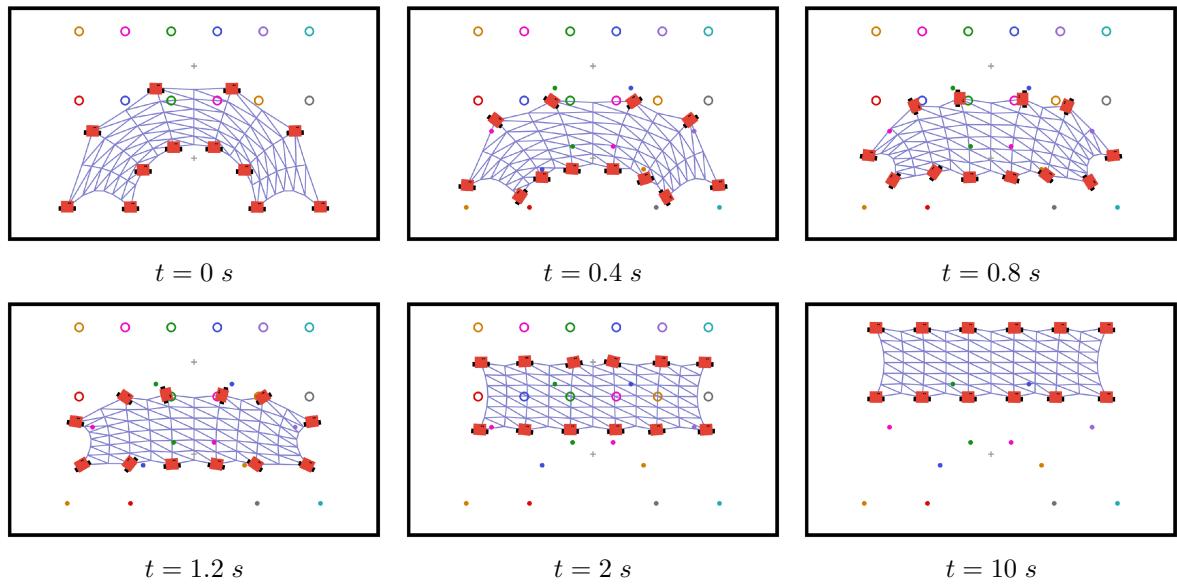


Figura C.1: Deformación de un objeto deformable durante su transporte en el Caso 1 ($k_G = 0$) del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular. Vídeo F.1.

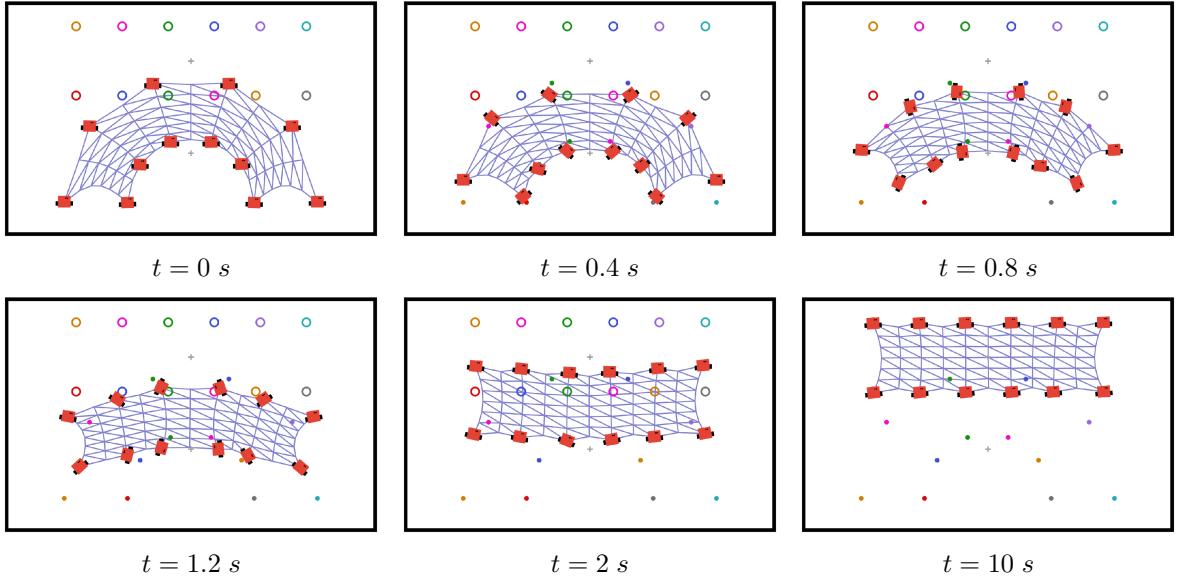


Figura C.2: Deformación de un objeto deformable durante su transporte en el Caso 2 ($k_G = 10$) del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular. Vídeo F.1.

El resto de figuras mostradas representan las trayectorias y posiciones de cada robot hasta llegar a su posición final, así como las velocidades lineales y angulares alcanzadas en cada instante de tiempo. En las figuras relativas a la velocidad se aprecia un salto brusco a valores nulos, entre los 4 y 6 segundos, consecuencia de la condición de parada definida para evitar la constante rotación de los robots en posiciones cercanas a la final.

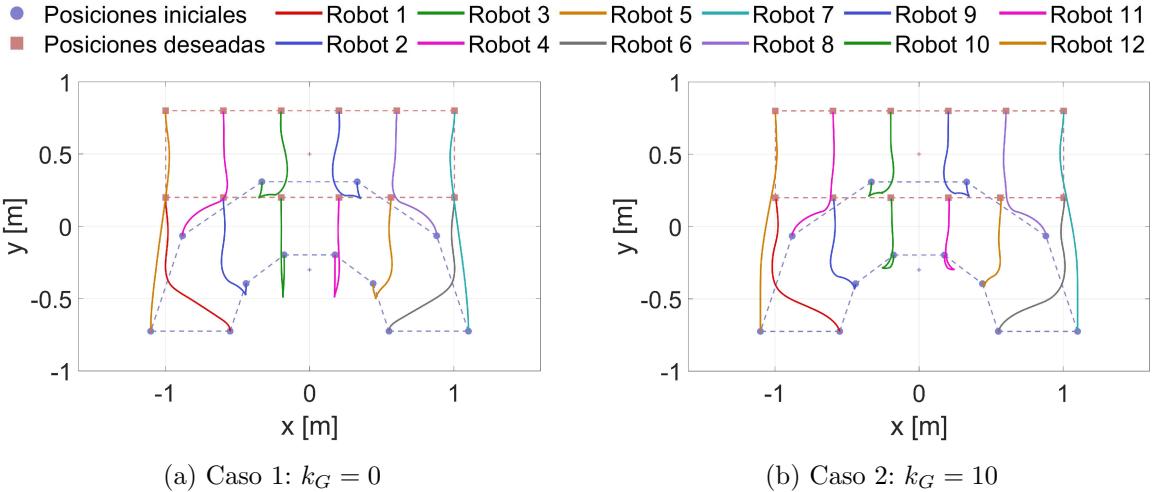


Figura C.3: Comparación de las trayectorias seguidas por los robots en los dos casos considerados del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular.

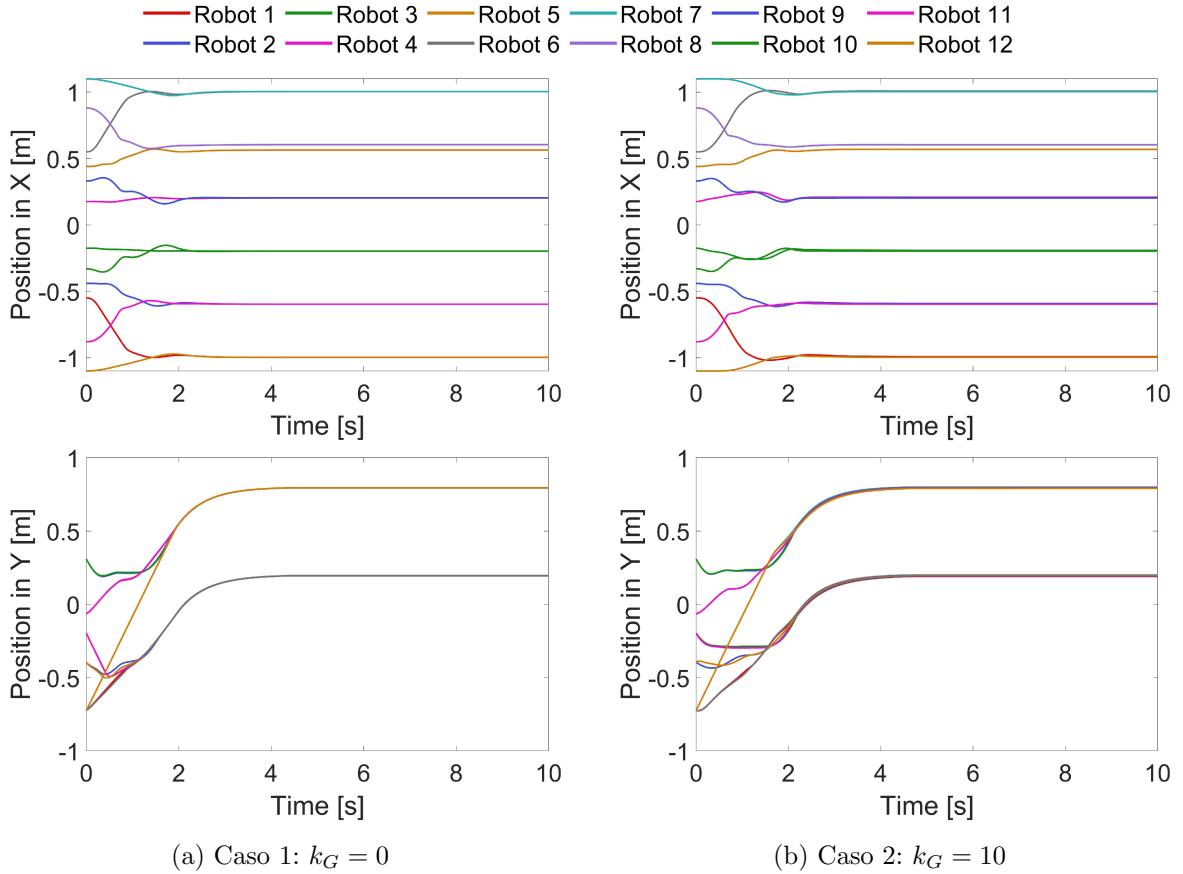


Figura C.4: Comparación de la variación de las coordenadas X e Y de posición en los dos casos considerados del experimento simulado donde el equipo de robots pasa de una configuración puramente dobrada a una configuración rectangular.

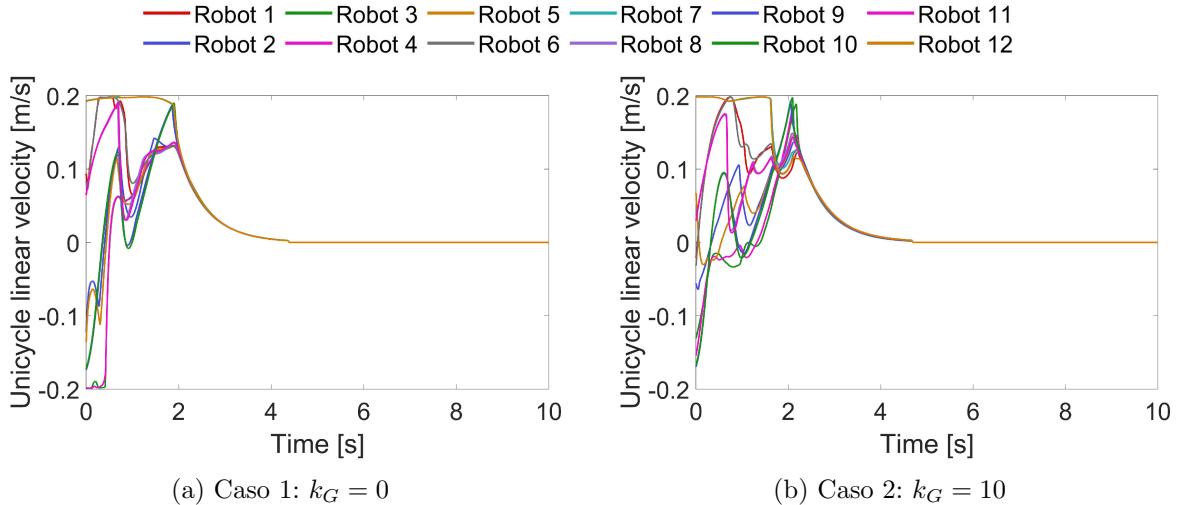


Figura C.5: Comparación de la variación de la velocidad lineal de uniciclo en los dos casos considerados del experimento simulado donde el equipo de robots pasa de una configuración puramente dobrada a una configuración rectangular.

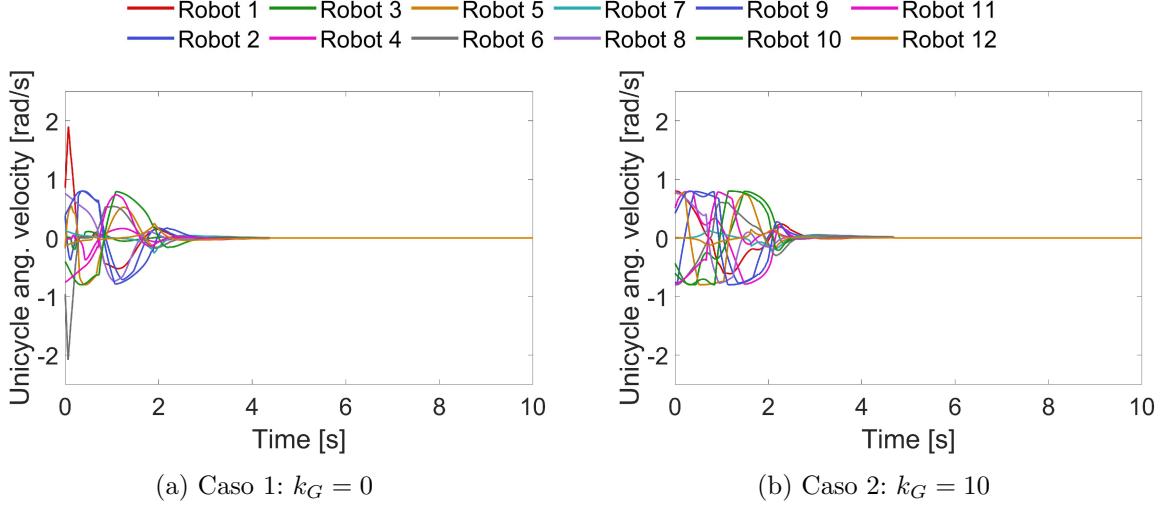


Figura C.6: Comparación de la variación de la velocidad angular de uniciclo en los dos casos considerados del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular.

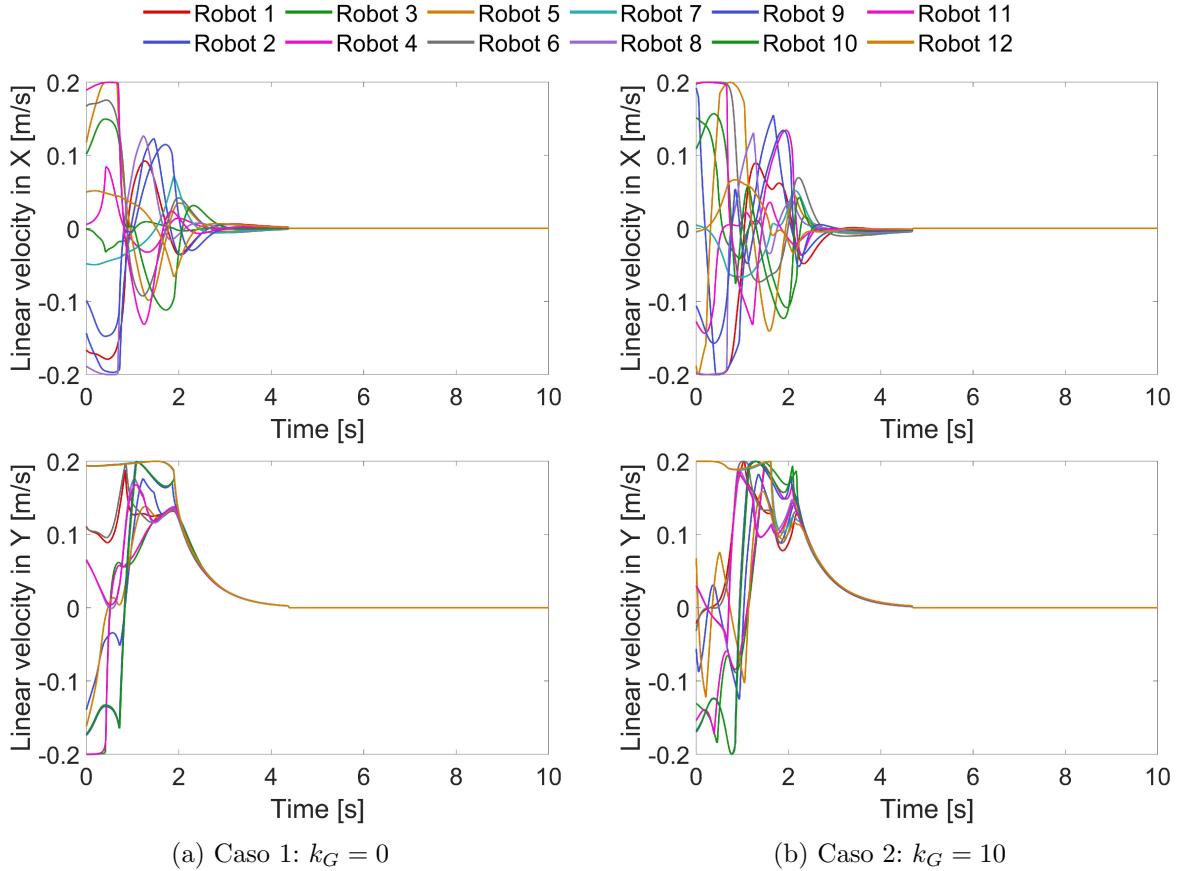


Figura C.7: Comparación de la variación de las componentes X e Y de velocidad lineal en los dos casos considerados del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular.

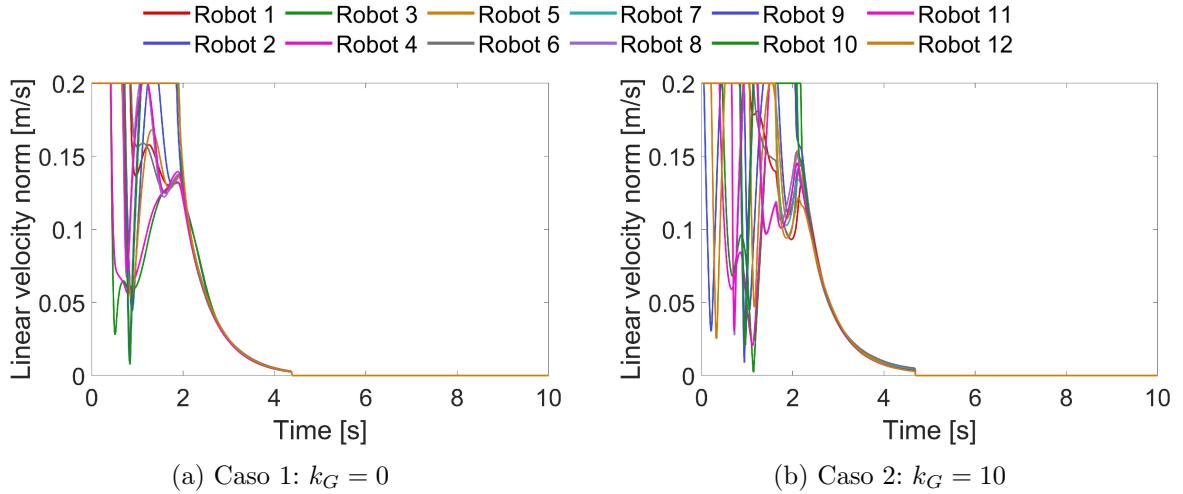


Figura C.8: Comparación de la variación del módulo de velocidad lineal en los dos casos considerados del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular.

C.2. Gráficas de la simulación 4.3

A continuación, se muestran resultados adicionales de la simulación de transporte de un objeto deformable en un entorno 2D con un obstáculo circular que impide el desplazamiento directo del equipo de robots hacia su posición final. En las Figuras C.9 y C.10, se muestran diferentes fotogramas de la tarea de transporte en los dos casos considerados. Se puede observar que las deformaciones sufridas en el Caso 2 están restringidas a modos de orden dos, frente a las deformaciones sin control del Caso 1.

Las Figuras C.11 y C.12 muestran las trayectorias seguidas por cada robot y sus coordenadas X e Y en cada instante de tiempo, respectivamente. En el Caso 1, las trayectorias son más directas, lo que influye en una mayor deformación del objeto.

El resto de figuras representan la evolución de diferentes variables de velocidad. Todas convergen a valores próximos a cero de forma adecuada. Debido a la condición de parada impuesta, las velocidades pasan a ser cero en torno a los 24 segundos en este ejemplo.

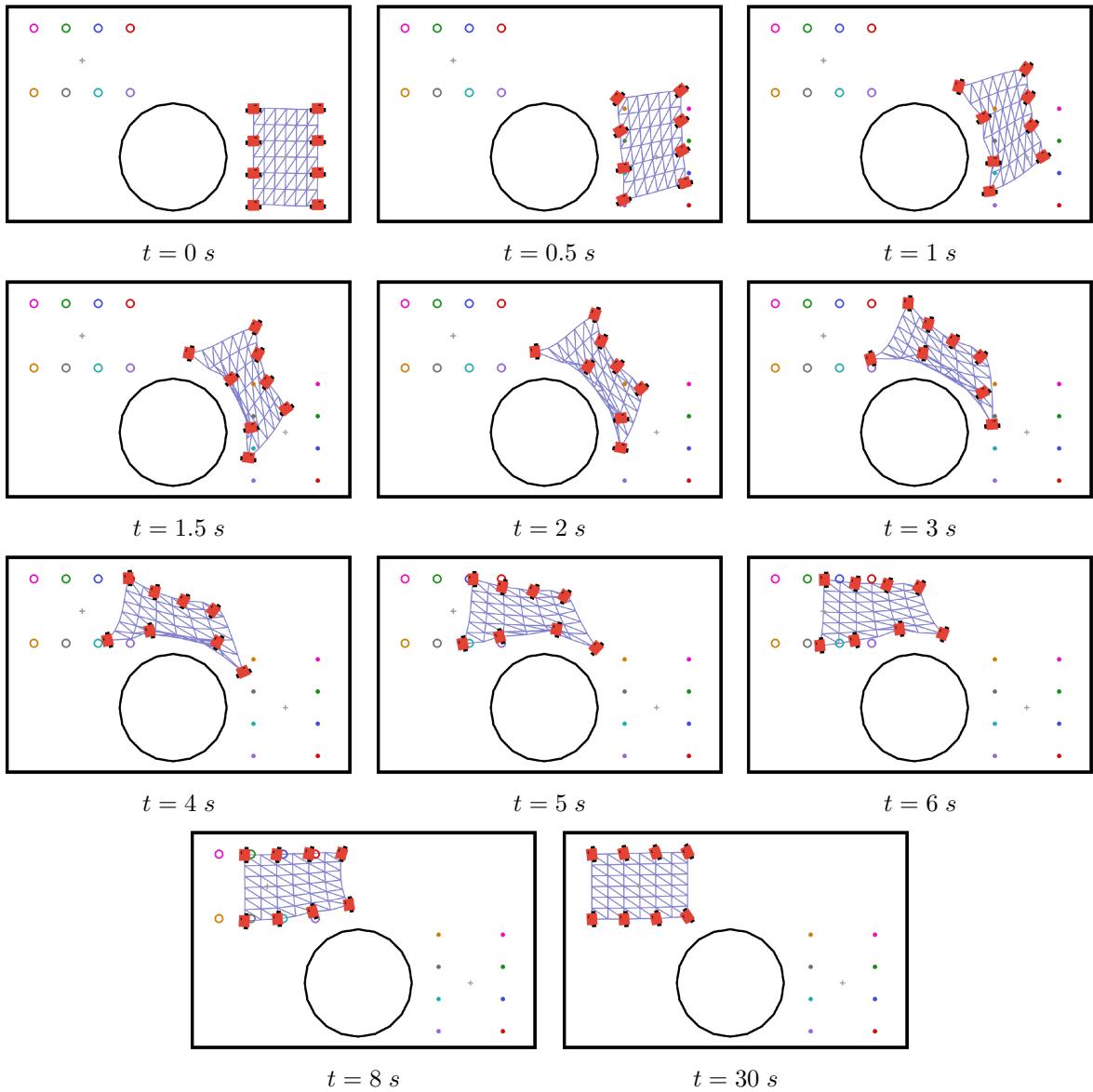


Figura C.9: Deformación de un objeto deformable durante su transporte en el Caso 1 ($k_G = 0$) del experimento simulado donde el equipo de robots debe esquivar un obstáculo circular. Vídeo F.2.

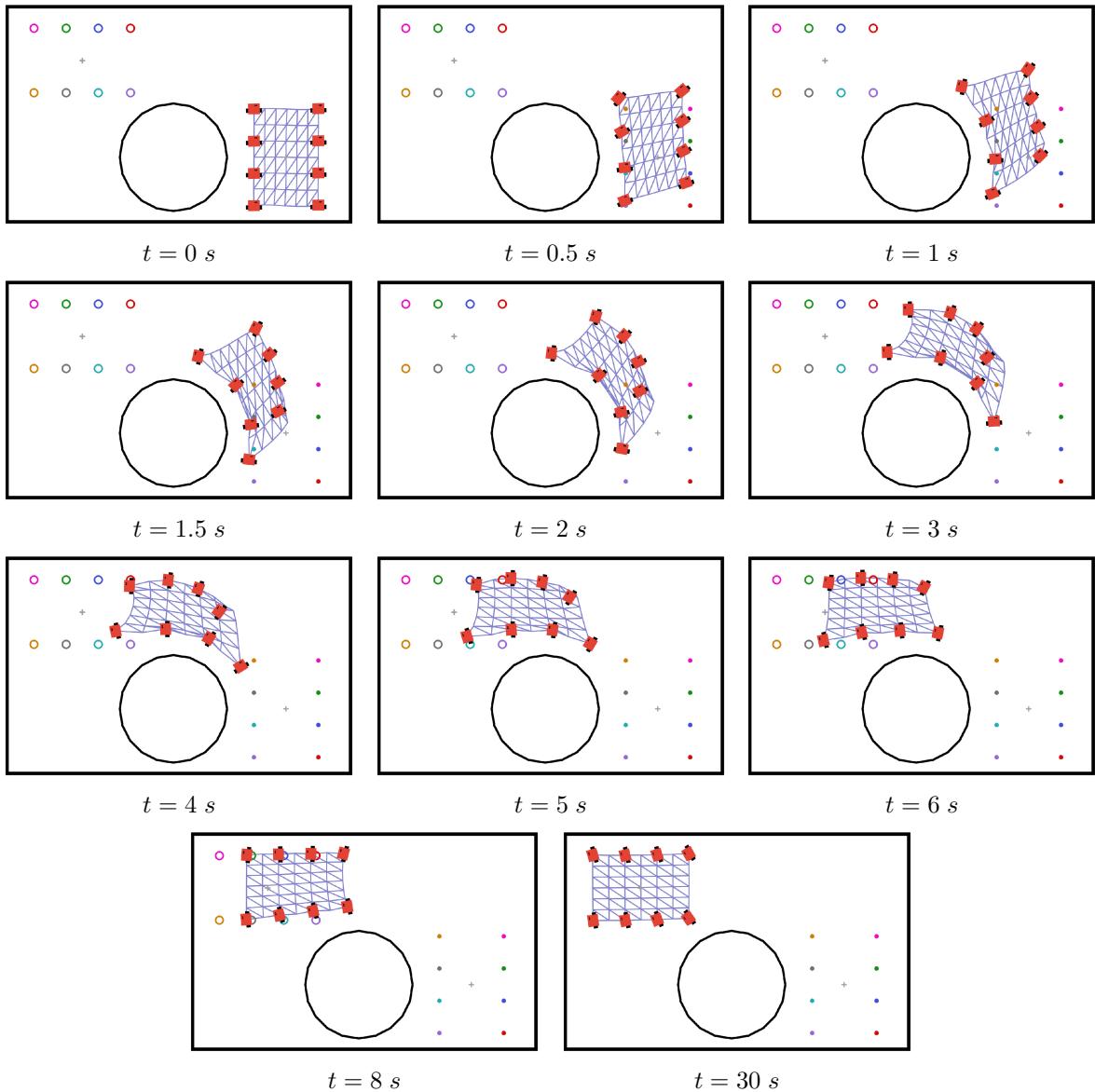


Figura C.10: Deformación de un objeto deformable durante su transporte en el Caso 2 ($k_G = 2$) del experimento simulado donde el equipo de robots debe esquivar un obstáculo circular. Vídeo F.2.

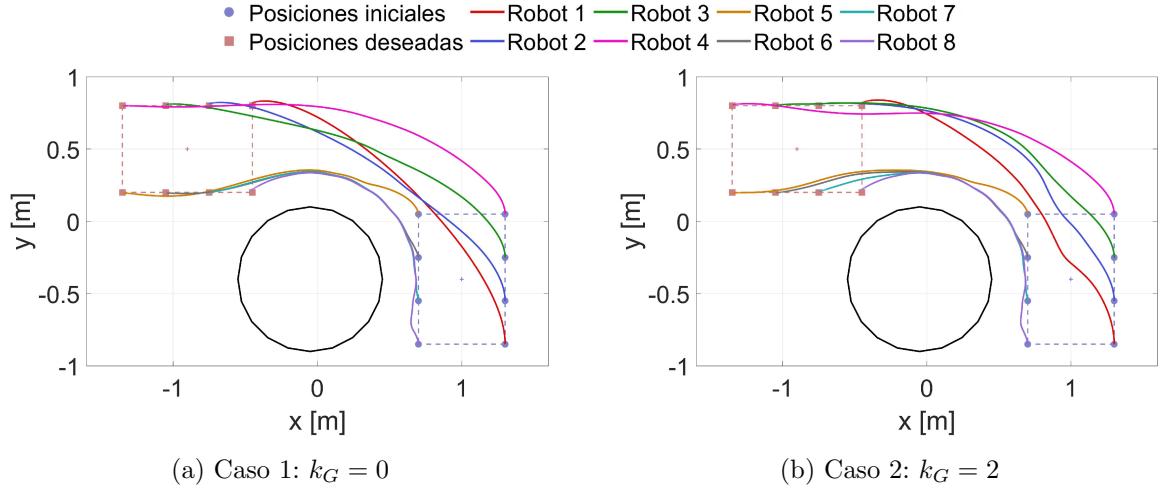


Figura C.11: Comparación de las trayectorias seguidas por los robots en los dos casos considerados del experimento simulado donde el equipo de robots debe esquivar un obstáculo circular.

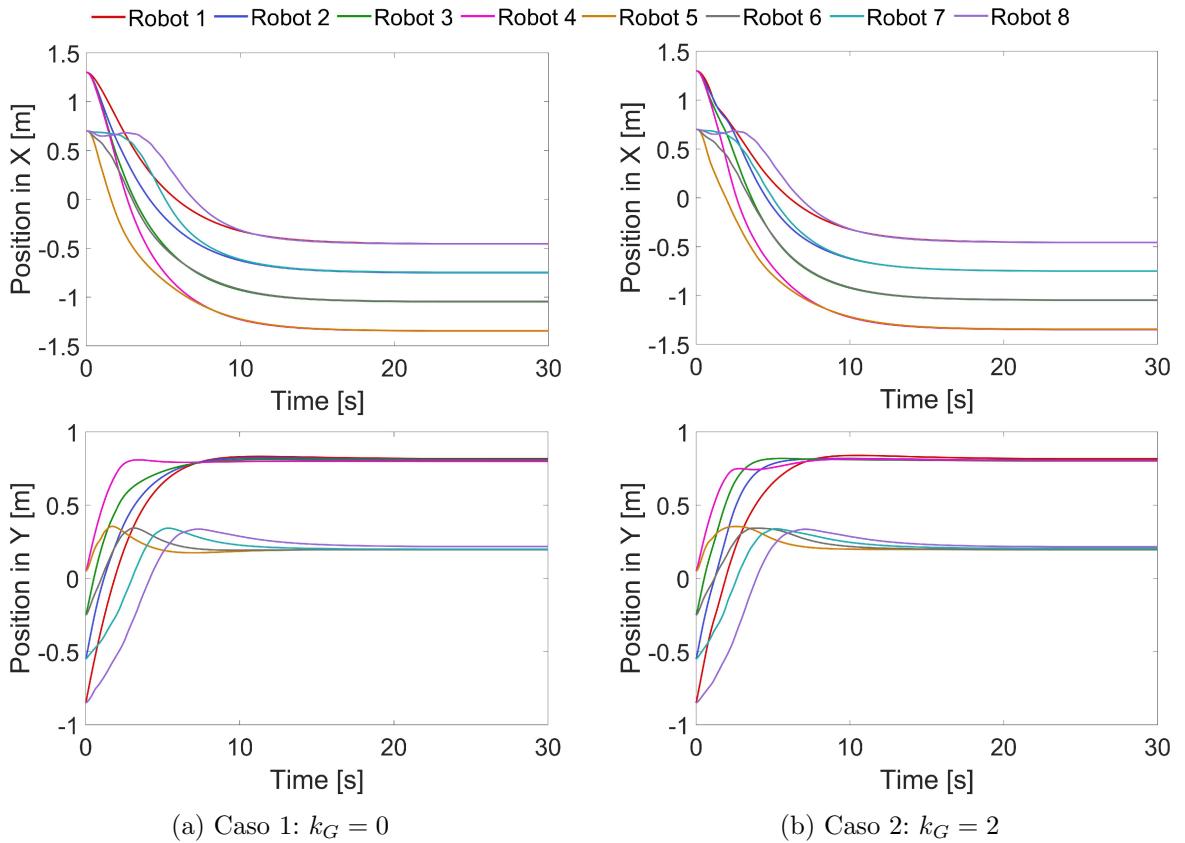


Figura C.12: Comparación de la variación de las coordenadas X e Y de posición en los dos casos considerados del experimento simulado donde el equipo de robots debe esquivar un obstáculo circular.

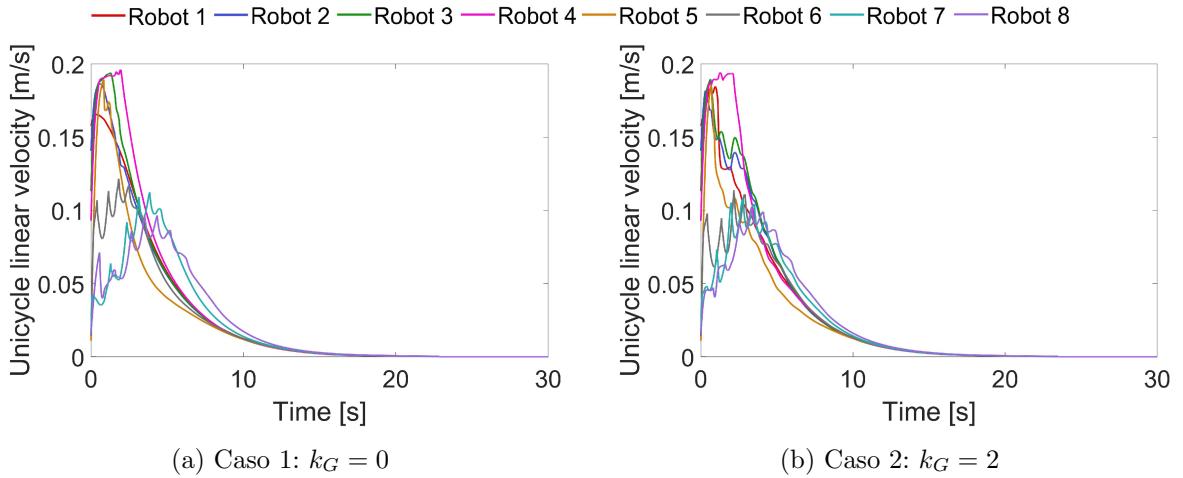


Figura C.13: Comparación de la variación de la velocidad lineal de uniciclo en los dos casos considerados del experimento simulado donde el equipo de robots debe esquivar un obstáculo circular.

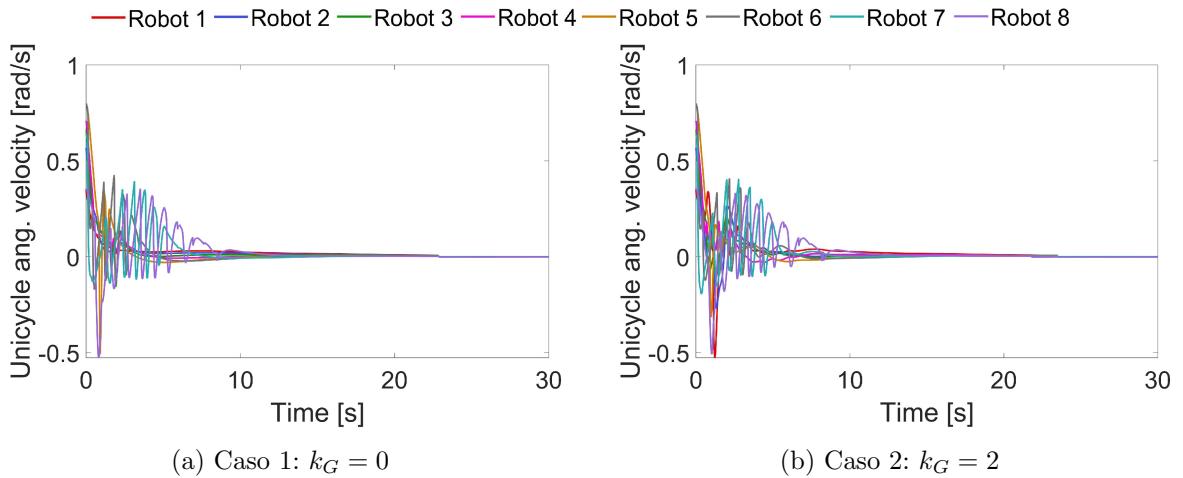


Figura C.14: Comparación de la variación de la velocidad angular de uniciclo en los dos casos considerados del experimento simulado donde el equipo de robots debe esquivar un obstáculo circular.

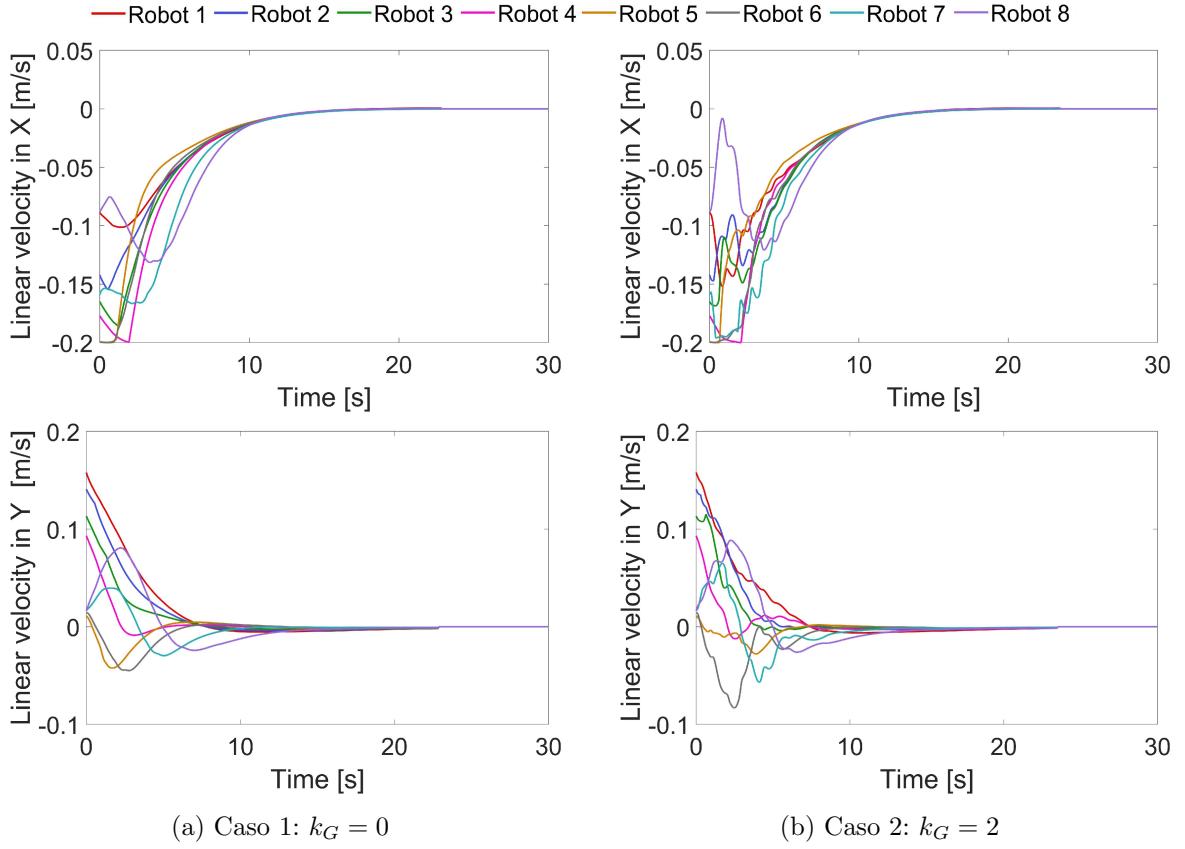


Figura C.15: Comparación de la variación de las componentes X e Y de velocidad lineal en los dos casos considerados del experimento simulado donde el equipo de robots debe esquivar un obstáculo circular.

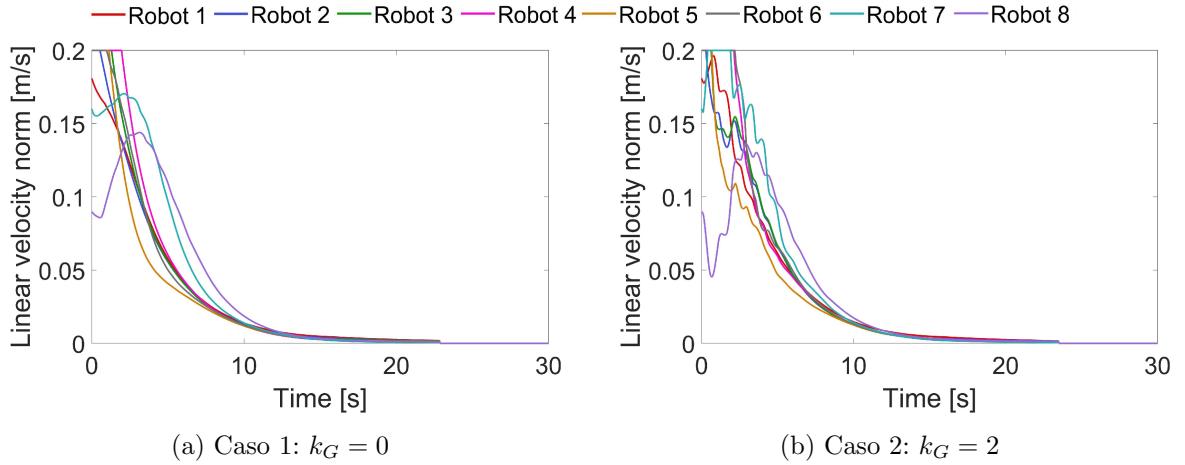


Figura C.16: Comparación de la variación del módulo de velocidad lineal en los dos casos considerados del experimento simulado donde el equipo de robots debe esquivar un obstáculo circular.

C.2.1. Conservación de forma durante el transporte

En las gráficas anteriores se ha visto que, en el Caso 1, la formación se deforma de manera no controlada al esquivar el obstáculo. Después, acaba convergiendo a la forma final deseada que, en este caso, coincide con la forma del objeto en reposo. También se puede considerar el caso en el que resulte imprescindible conservar la forma del objeto durante todo el transporte. A continuación, se muestran los resultados más relevantes obtenidos al simular esta situación, con los siguientes valores de ganancias de control: $k_H = 50$, $k_G = 2$, $k_s = 0.15$, $k_c = 0.1$ y $k_\theta = 0.15$.

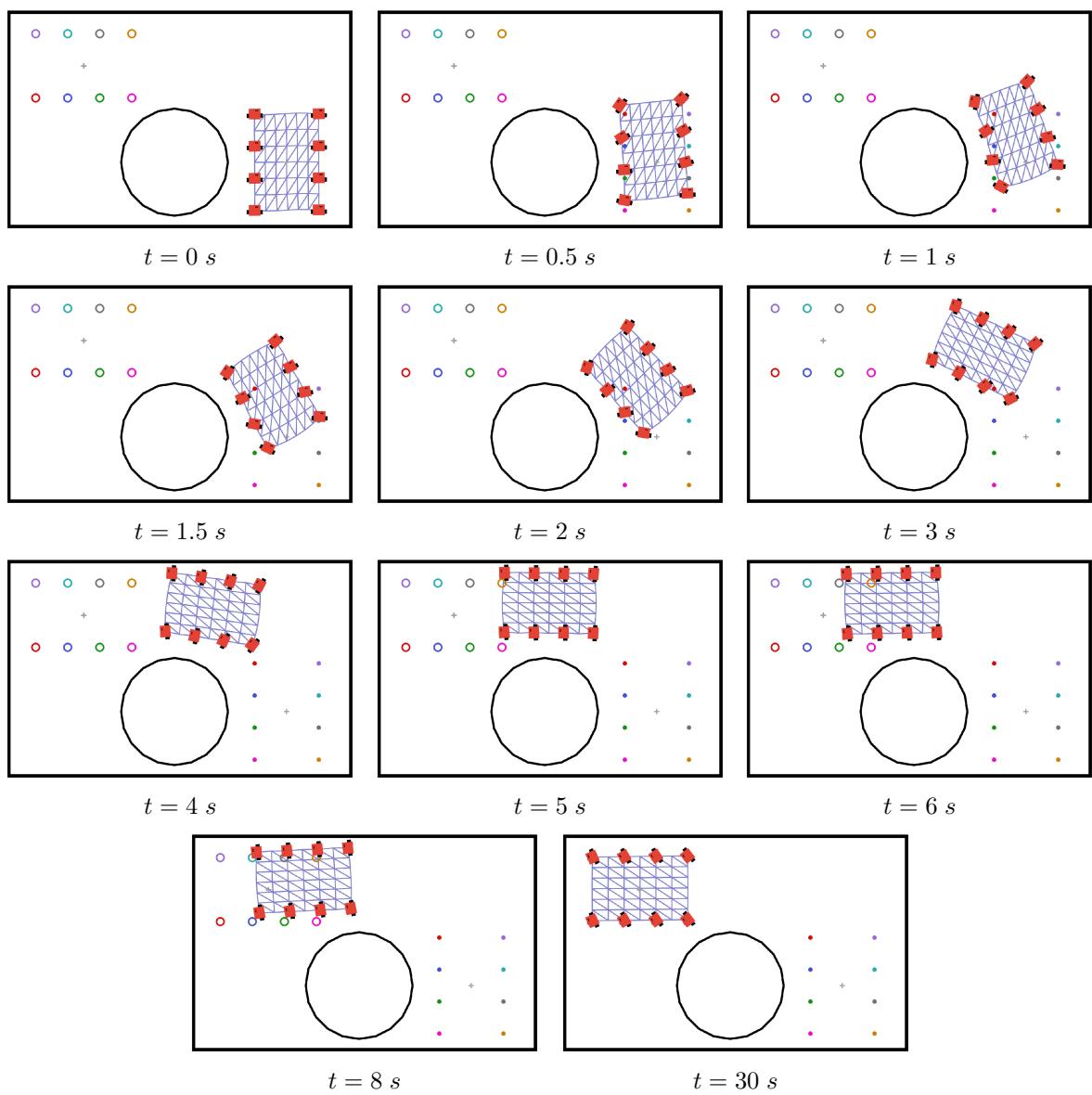


Figura C.17: Conservación de forma de un objeto deformable durante su transporte ($k_H = 50$) en el experimento simulado donde el equipo de robots debe esquivar un obstáculo circular. Vídeo F.2.1.

En la Figura C.18, se puede comprobar que el error de forma se ha reducido a valores muy próximos a cero y que las trayectorias siguen un contorno prácticamente circular, evitando que el objeto se deforme.

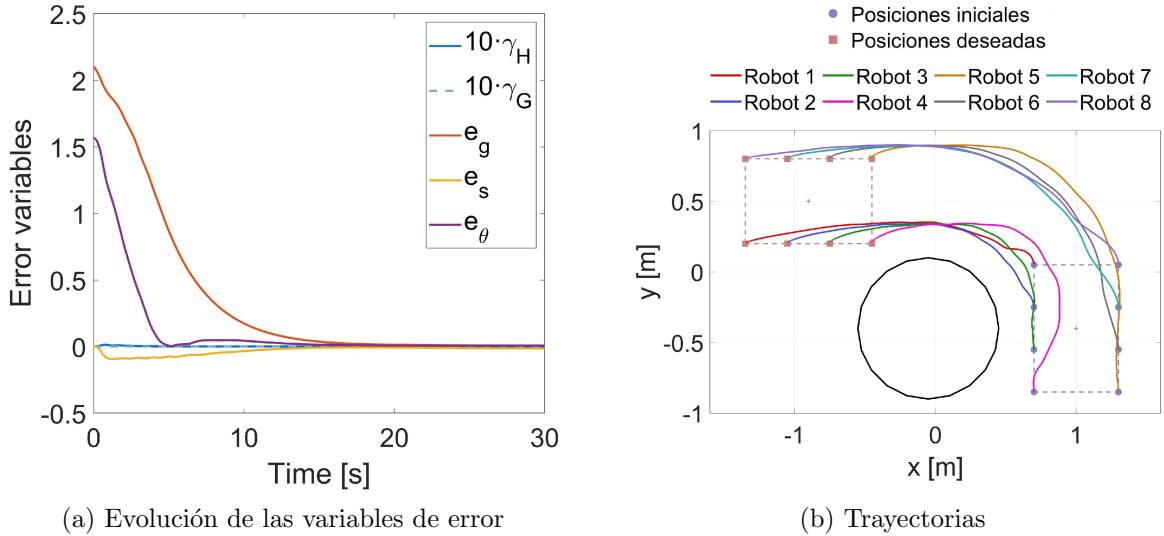


Figura C.18: Representación de la evolución de las variables de error y las trayectorias seguidas por los robots en el experimento simulado donde el equipo de robots debe esquivar un obstáculo circular con $k_H = 50$.

C.3. Gráficas del experimento 4.4

En este apartado se recogen los resultados gráficos obtenidos en el simulador y en la plataforma experimental con robots reales del experimento donde el equipo de robots transporta un objeto deformable a lo largo de un pasillo curvo. Esto permite comparar las diferencias y similitudes entre simulación y experimento real.

C.3.1. Resultados obtenidos en simulación

A continuación, se muestran los resultados de las simulaciones realizadas en los tres casos considerados en este experimento. En las Figuras C.19, C.20 y C.21 se aprecia la deformación que sufre el objeto durante su transporte, pudiéndose comprobar que en la peor situación, el Caso 0, la formación se deforma sin control y llega a la posición final sin recuperar su forma original. En el Caso 1 se nota cierta variación en γ_G , mientras que en el Caso 2 es constante y nula. Todo esto también se puede ver en las gráficas de la evolución de errores de la Figura C.22.

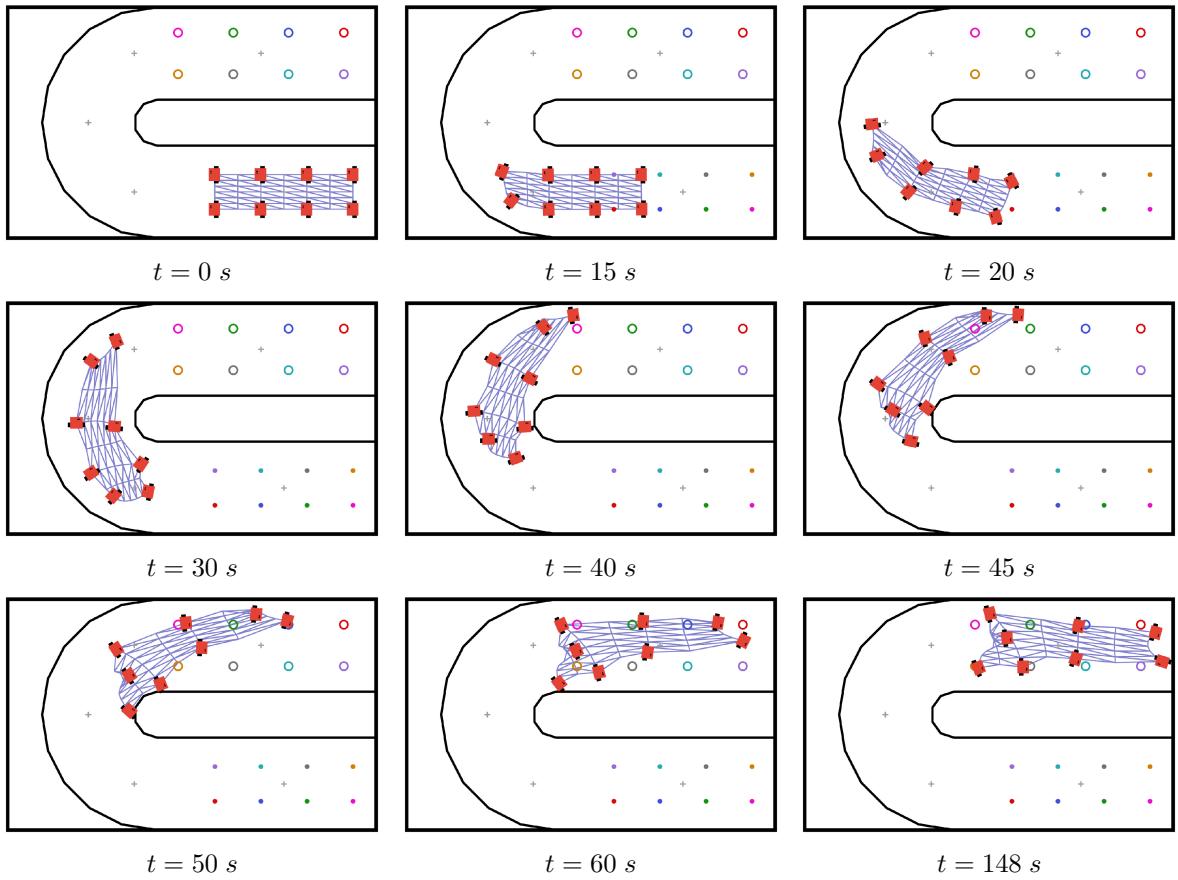


Figura C.19: Deformación de un objeto deformable durante su transporte en el Caso 0 ($k_H = 0$ y $k_G = 0$) del experimento simulado donde el equipo de robots recorre un pasillo curvo en un espacio 2D. Vídeo F.3.1.

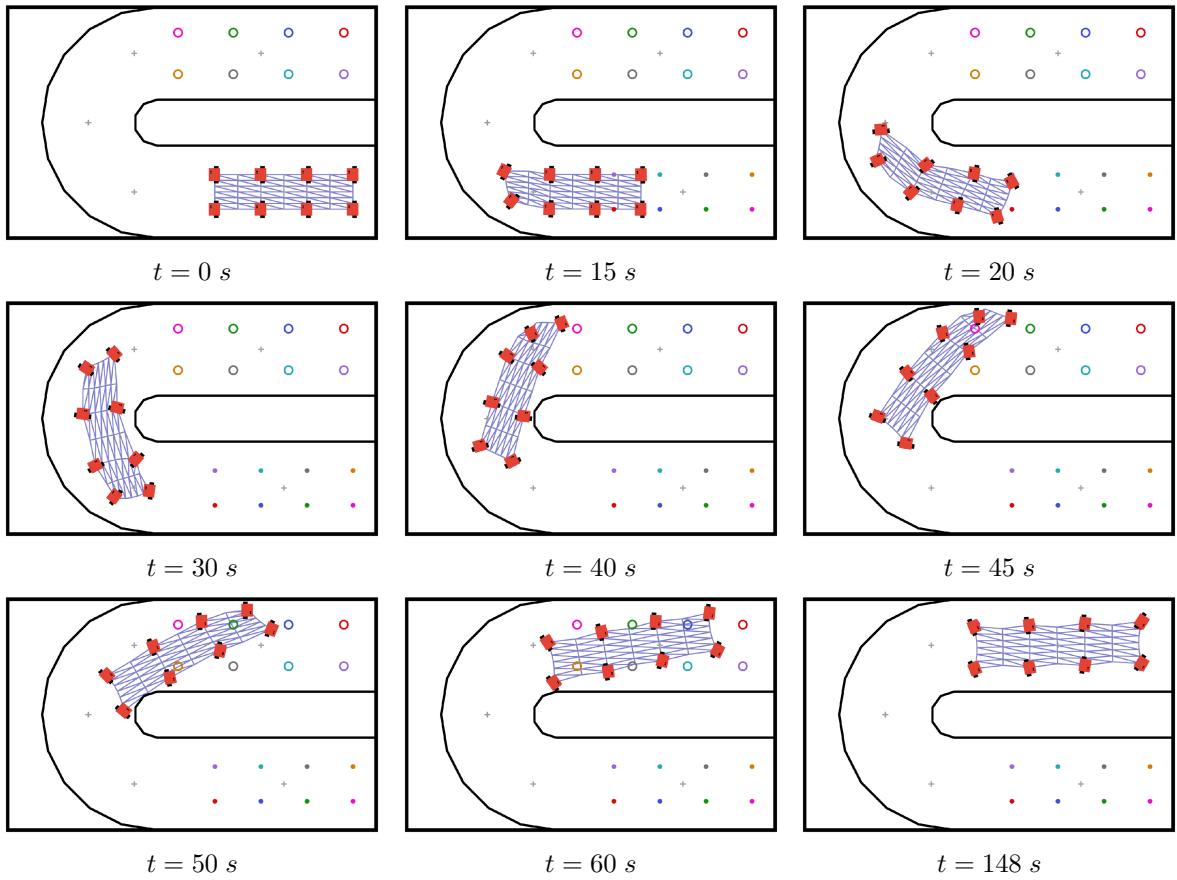


Figura C.20: Deformación de un objeto deformable durante su transporte en el Caso 1 ($k_H = 0.2$ y $k_G = 0$) del experimento simulado donde el equipo de robots recorre un pasillo curvo en un espacio 2D. Vídeo F.3.1.

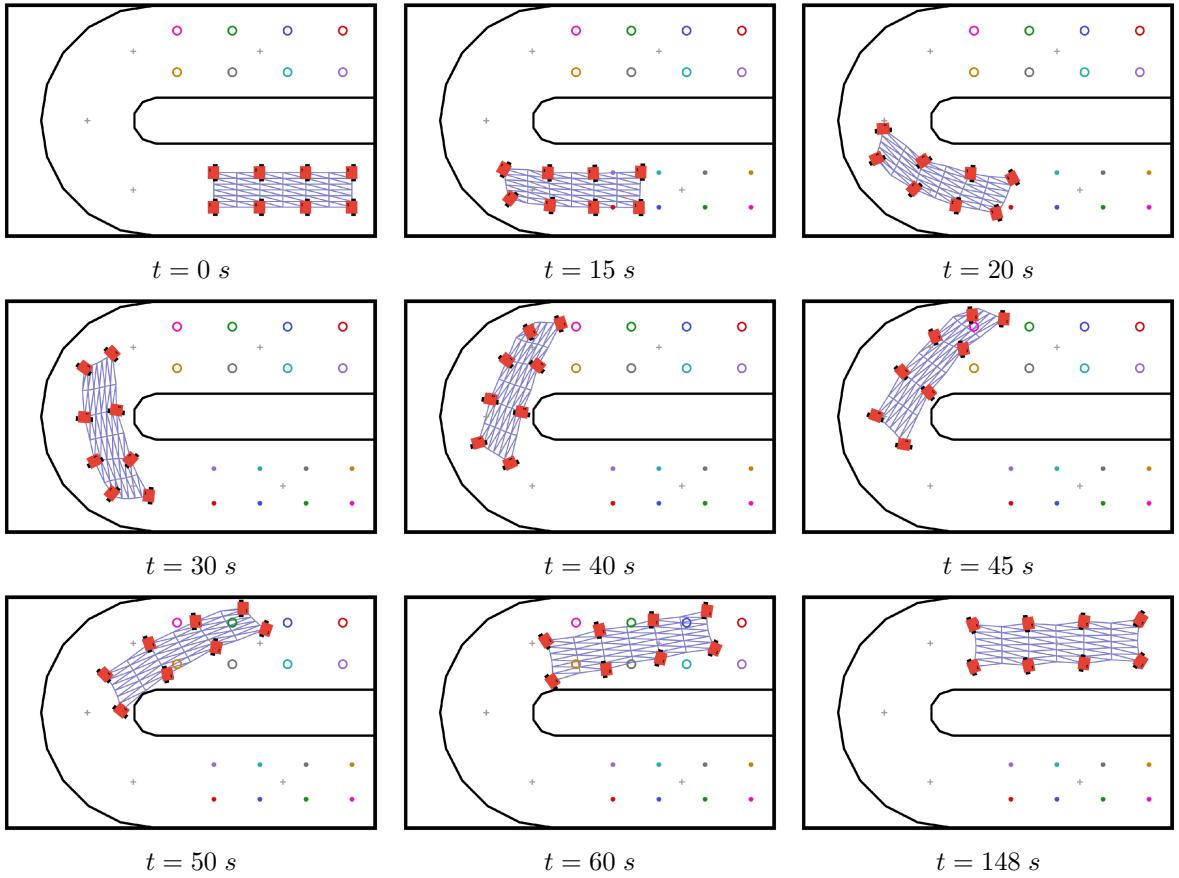


Figura C.21: Deformación de un objeto deformable durante su transporte en el Caso 2 ($k_H = 0.2$ y $k_G = 1$) del experimento simulado donde el equipo de robots recorre un pasillo curvo en un espacio 2D. Vídeo F.3.1.

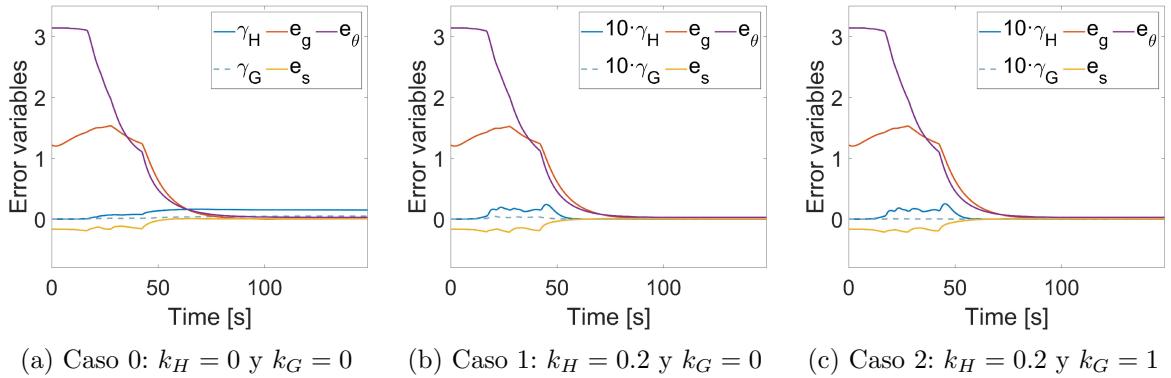


Figura C.22: Comparación de la evolución de las variables de error en los tres casos considerados del experimento simulado donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

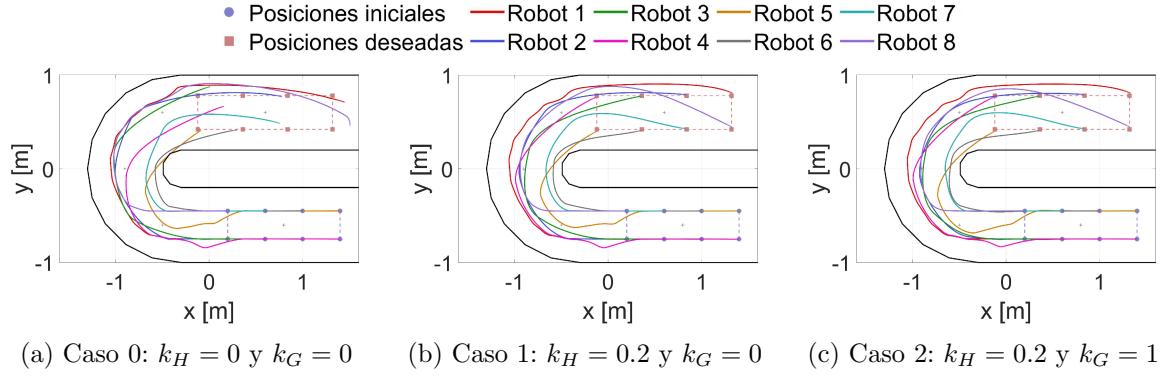


Figura C.23: Comparación de las trayectorias seguidas por los robots en los tres casos considerados del experimento simulado donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

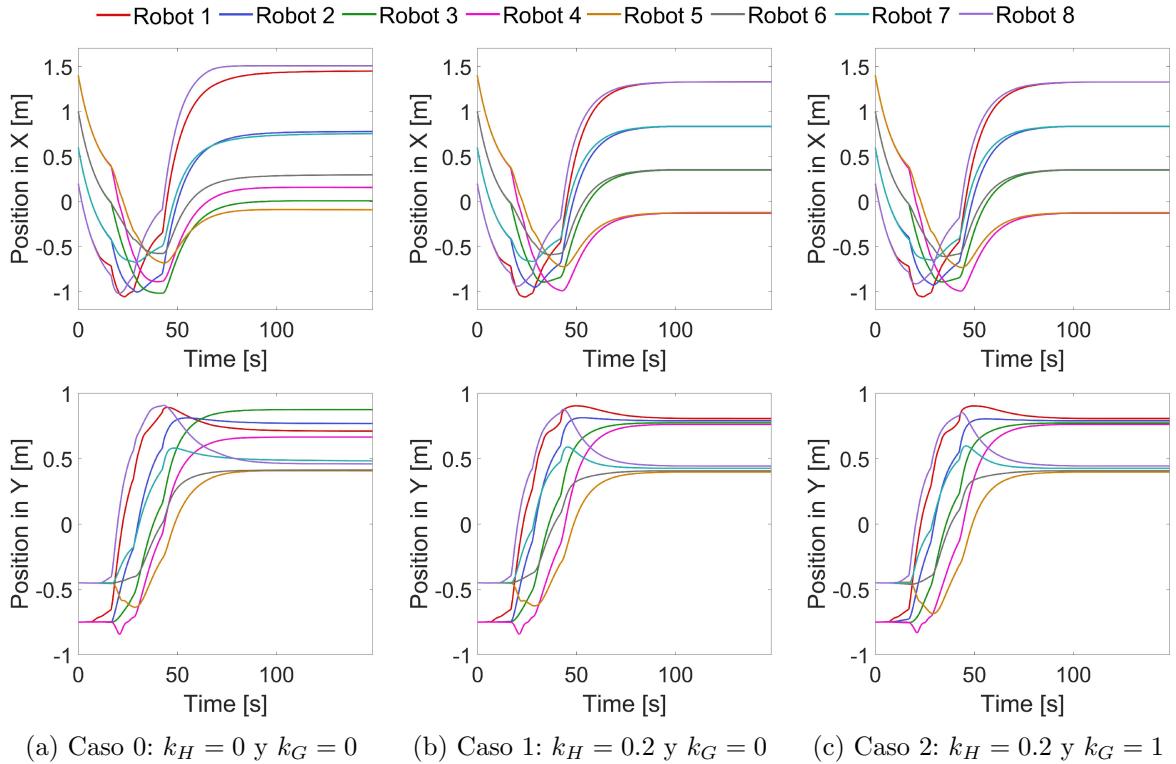


Figura C.24: Comparación de la variación de las coordenadas X e Y de posición en los tres casos considerados del experimento simulado donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

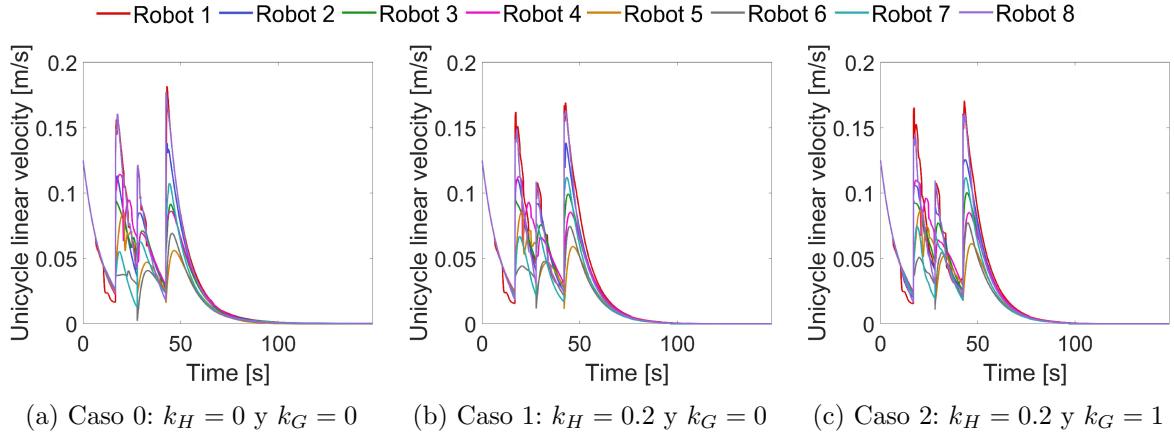


Figura C.25: Comparación de la variación de la velocidad lineal de uniciclo en los tres casos considerados del experimento simulado donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

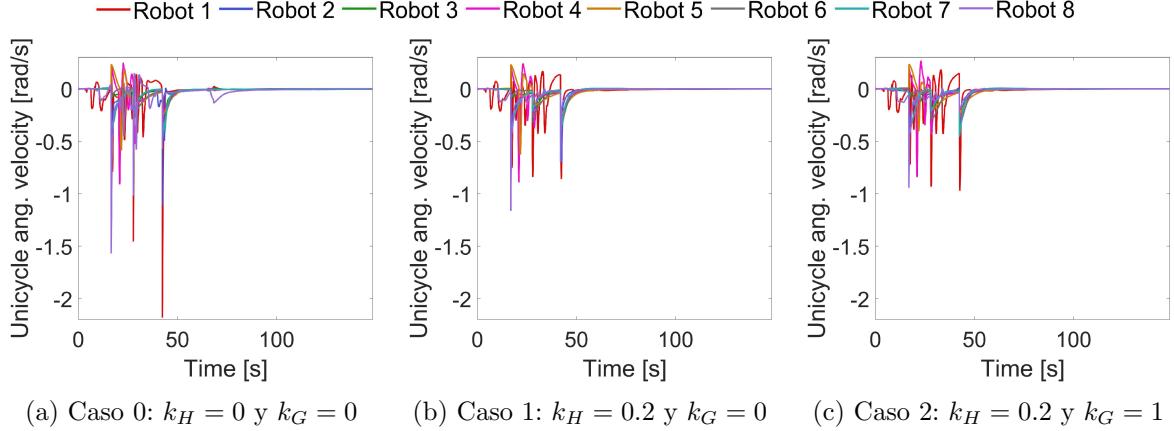


Figura C.26: Comparación de la variación de la velocidad angular de uniciclo en los tres casos considerados del experimento simulado donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

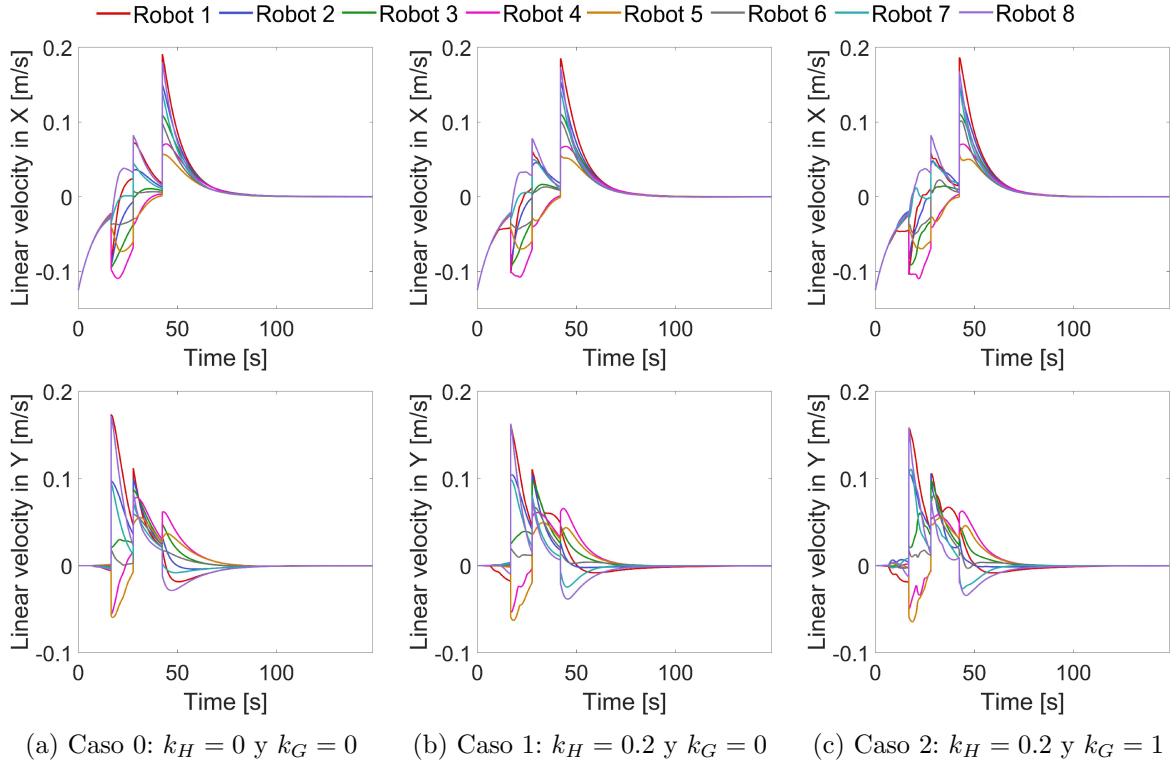


Figura C.27: Comparación de la variación de las componentes X e Y de velocidad lineal en los tres casos considerados del experimento simulado donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

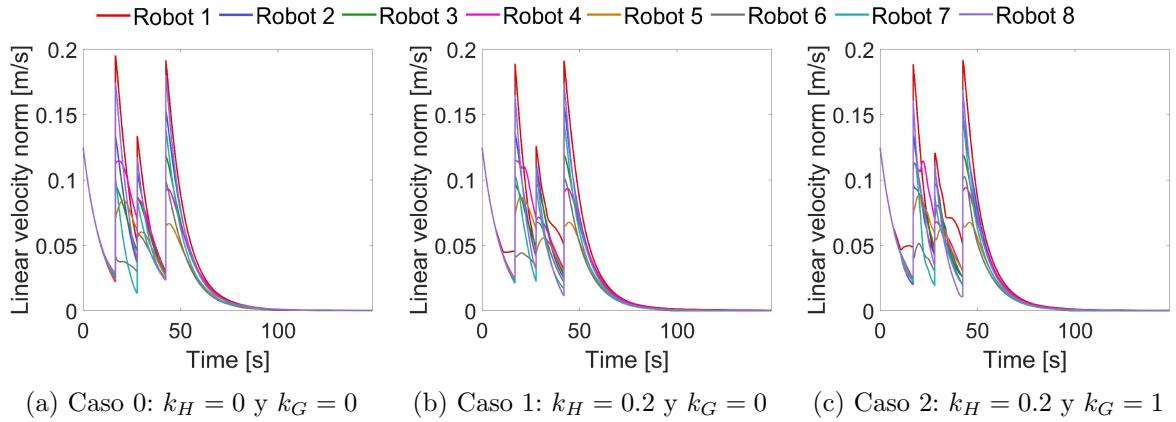


Figura C.28: Comparación de la variación del módulo de velocidad lineal en los tres casos considerados del experimento simulado donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

C.3.2. Resultados obtenidos con robots reales

Las siguientes gráficas recogen los resultados restantes obtenidos tras la ejecución de este experimento con robots reales.

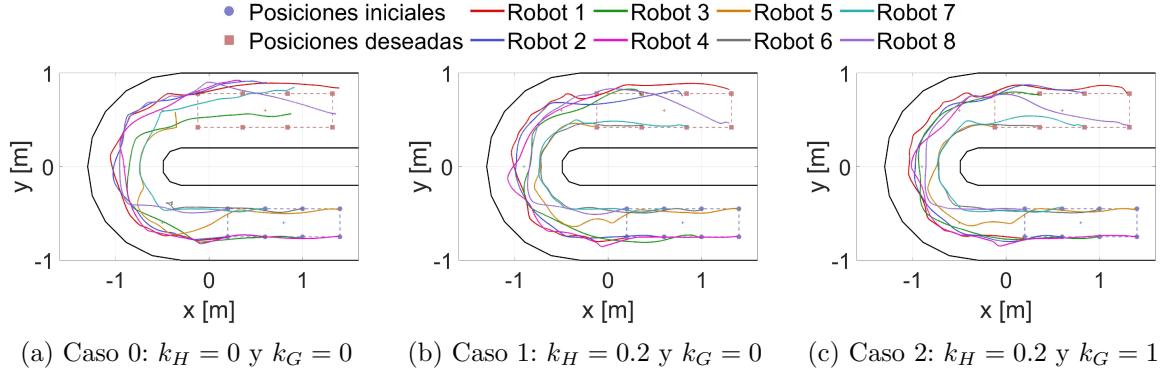


Figura C.29: Comparación de las trayectorias seguidas por los robots en los tres casos considerados del experimento donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

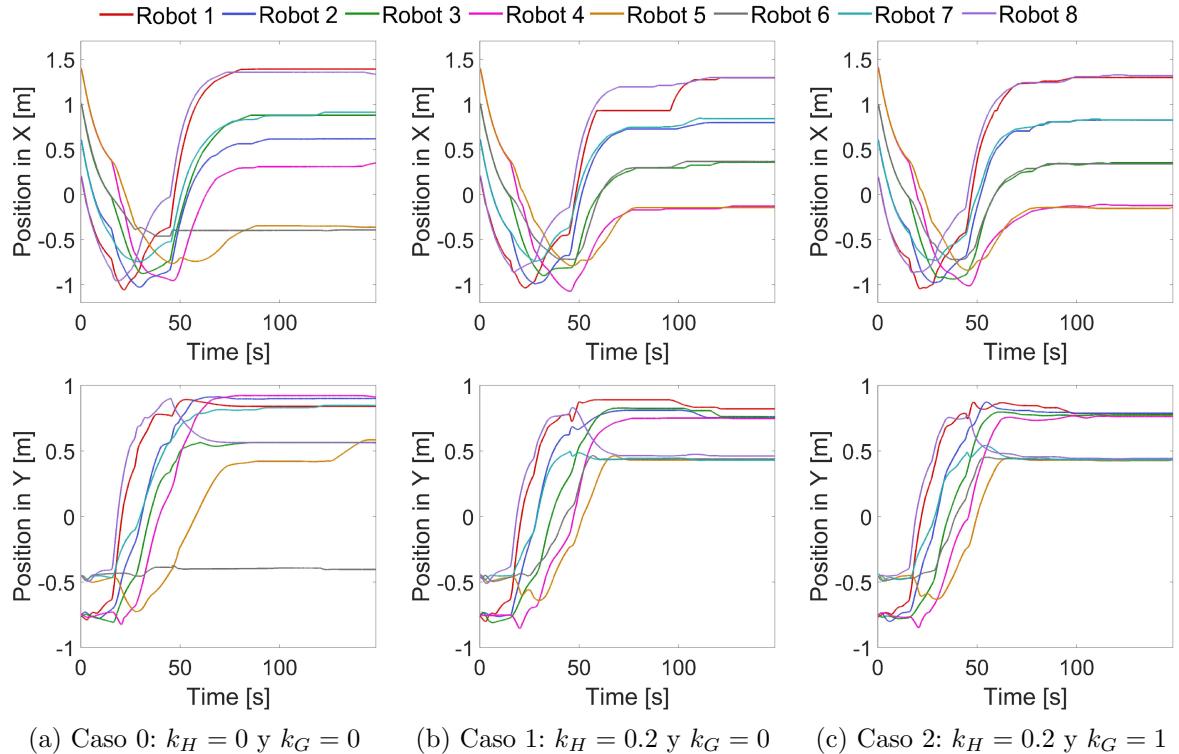


Figura C.30: Comparación de la variación de las coordenadas X e Y de posición en los tres casos considerados del experimento donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

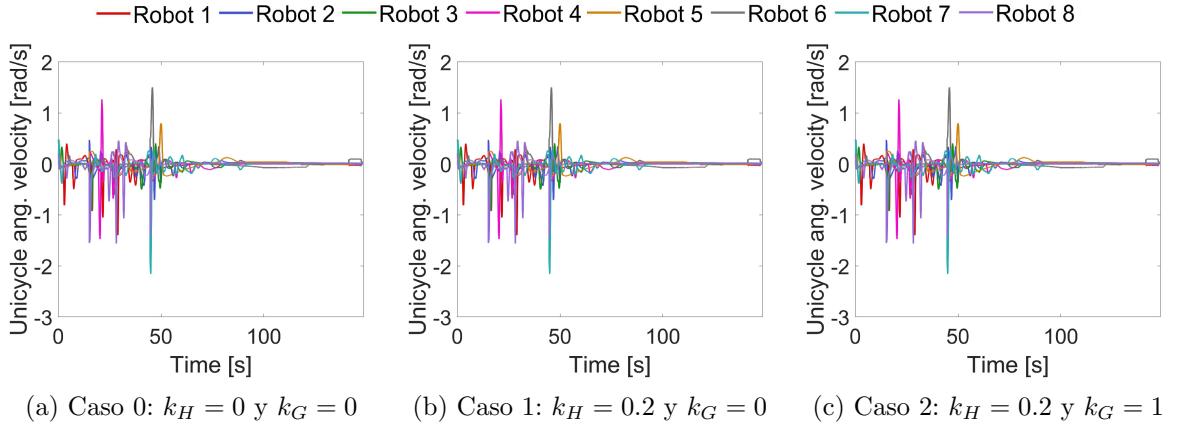


Figura C.31: Comparación de la variación de la velocidad angular de uniciclo en los tres casos considerados del experimento donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

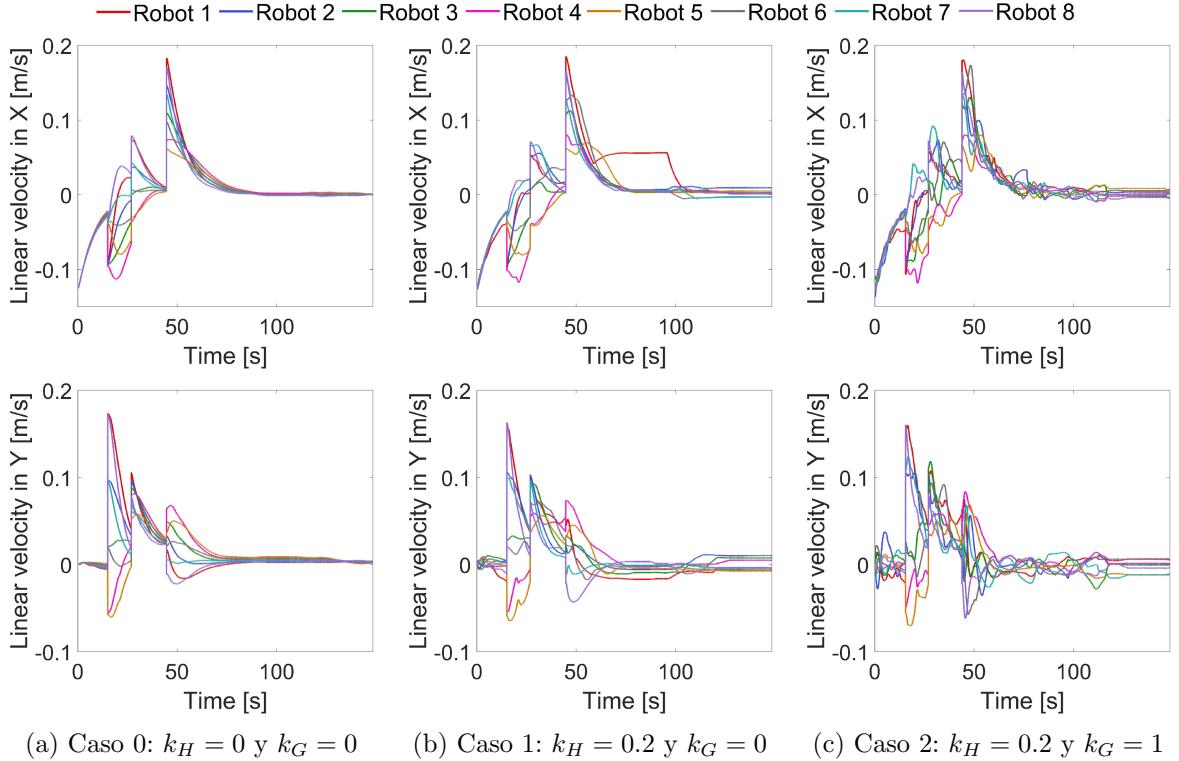


Figura C.32: Comparación de la variación de las componentes X e Y de velocidad lineal en los tres casos considerados del experimento donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

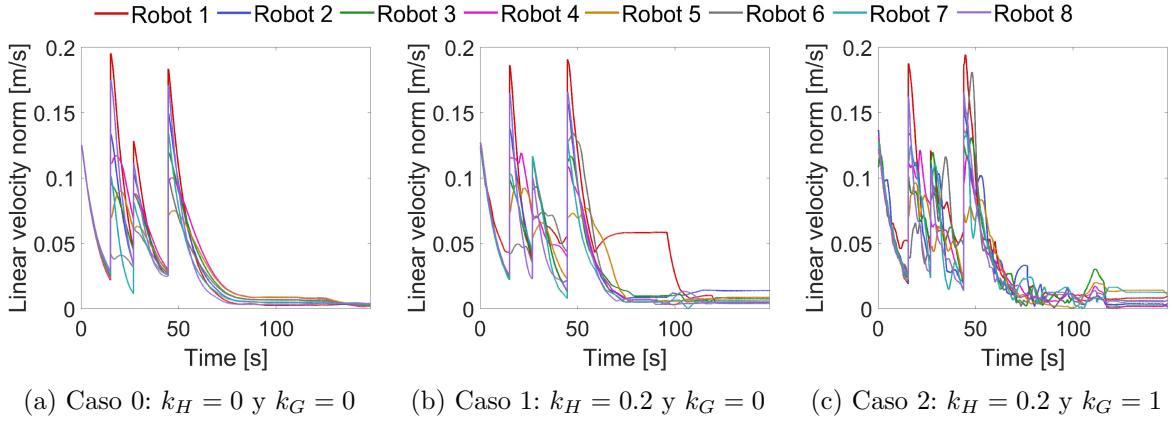


Figura C.33: Comparación de la variación del módulo de velocidad lineal en los tres casos considerados del experimento donde el equipo de robots recorre un pasillo curvo en un espacio 2D.

C.3.3. Justificación de la necesidad de *waypoints*

Este experimento ha llegado a buen término gracias a que se han definido una serie de puntos intermedios que marcasen la ruta que debía seguir el equipo de robots. Para justificar la necesidad de incorporar estos *waypoints*, en la Figura C.34 se muestran algunos fotogramas del movimiento que realizan los robots cuando no se planifica previamente su trayectoria. Se puede observar que la formación intenta rotar y desplazarse en un espacio limitado para alcanzar la configuración deseada, llegando a una situación de bloqueo.

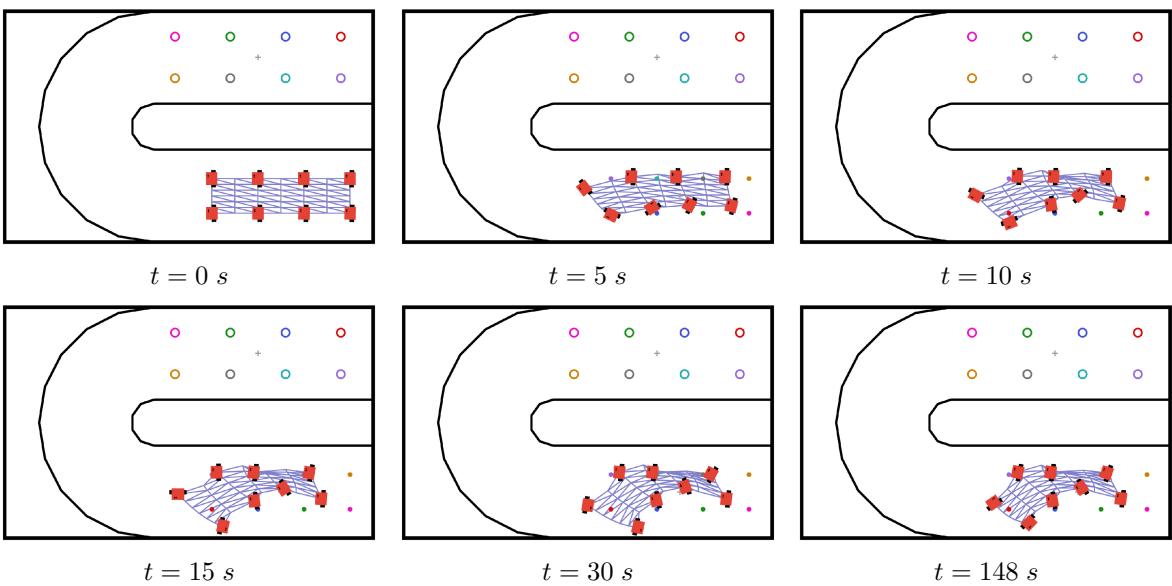


Figura C.34: Deformación de un objeto deformable durante su transporte en el experimento simulado donde el equipo de robots recorre un pasillo curvo sin planificación de la trayectoria en un espacio 2D.

C.4. Transporte con diferente número de robots

En este apartado se incluye un experimento adicional en el que un equipo de robots transporta un objeto deformable desde una configuración deformada hasta una configuración rectangular en un espacio bidimensional. El objetivo es comparar la influencia del número de robots involucrado en el transporte. Además, se usará uno de estos ejemplos para demostrar que el controlador implementado está desacoplado. Las ganancias de control son: $k_H = 0.1$, $k_G = 2$, $k_s = 0.15$, $k_c = 0.08$ y $k_\theta = 0.15$.

Cabe destacar que en los siguientes ejemplos no se ha corregido la posición final de la formación. Por tanto, como consecuencia de la condición de parada, el error de posición no convergerá a cero.

A partir de las gráficas de deformación, evolución de errores y trayectorias mostradas a continuación, se puede comprobar que solo tiene sentido aplicar el controlador con modos de deformación cuando el número de robots encargados de transportar el objeto deformable es, como mínimo, seis.

Cuando la tarea se lleva a cabo por 4 robots, la variable γ_G es constante e igual a cero durante todo el experimento. Esto es porque cualquier deformación que sufra la formación será siempre una combinación de transformaciones lineales. En ningún caso podrá alcanzar una configuración *bend*. En cambio, con seis o más robots, el objeto sí puede sufrir deformaciones cuadráticas y, por tanto, γ_G varía.

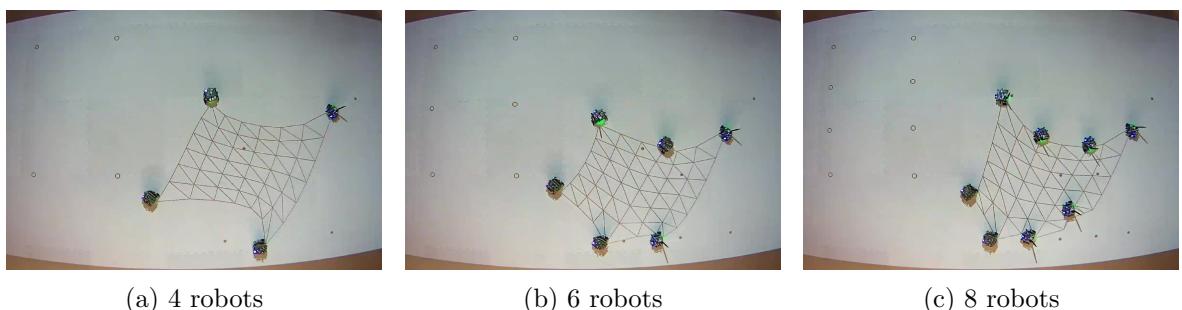


Figura C.35: Deformación de un objeto deformable en el instante $t = 4$ s de su transporte con diferente número de robots en un espacio 2D. Vídeo F.4.

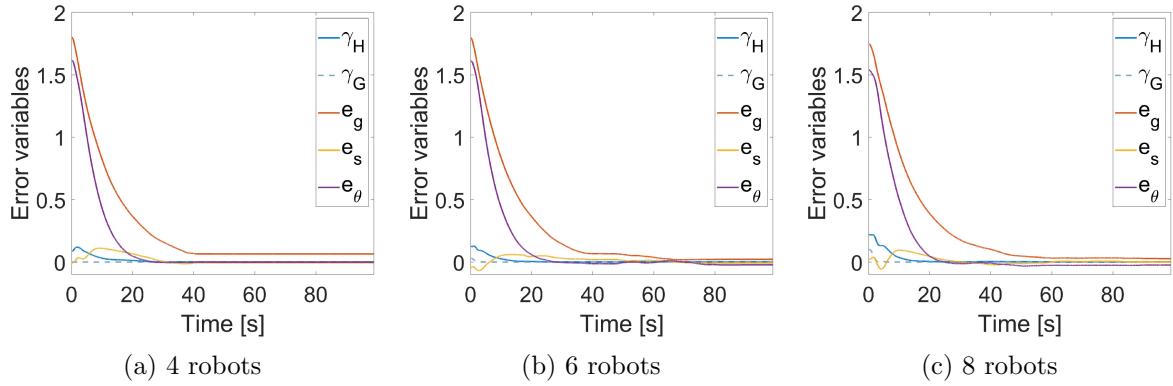


Figura C.36: Comparación de la evolución de las variables de error del experimento donde un equipo de diferente número de robots transporta un objeto deformable en un espacio 2D.

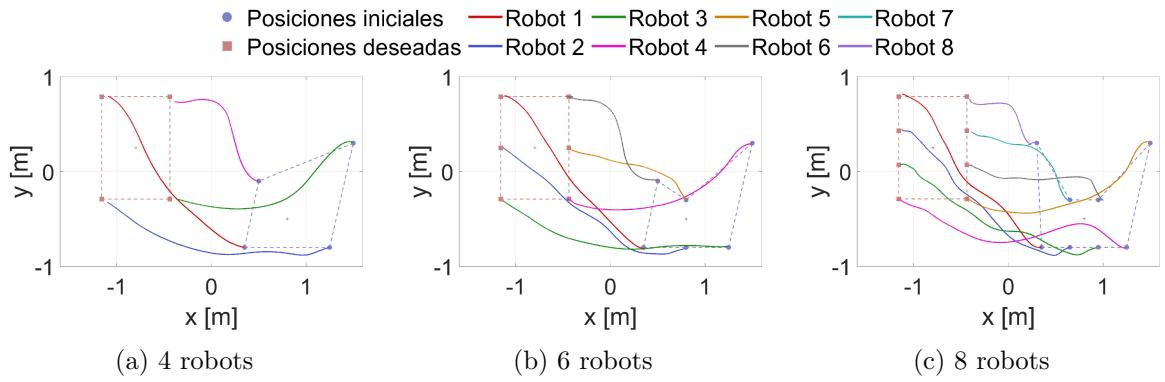


Figura C.37: Comparación de las trayectorias del experimento donde un equipo de diferente número de robots transporta un objeto deformable en un espacio 2D.

C.4.1. Demostración de control de escala desacoplado

Como se ha visto en la Figura C.36, el error de escala sufre cierta oscilación en los tres casos. Esta variación se puede reducir o, incluso, eliminar aumentando el peso de la ganancia de control relativa a la escala. Para demostrar esto, se propone el mismo ejemplo de antes con 8 robots, pero tomando $k_s = 2$.

Comparando la Figura C.38 con la evolución de los errores del apartado anterior se observa que el error de escala se ha reducido prácticamente por completo sin apenas afectar al resto de variables.

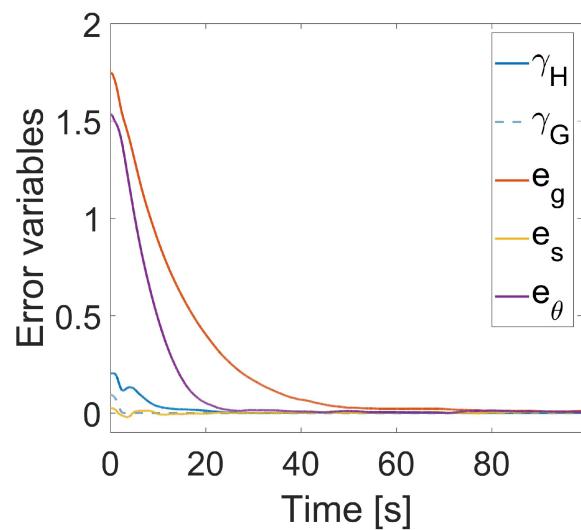


Figura C.38: Evolución de las variables de error del experimento donde un equipo de 8 robots transporta un objeto deformable en un espacio 2D cuando se quiere conservar la escala.

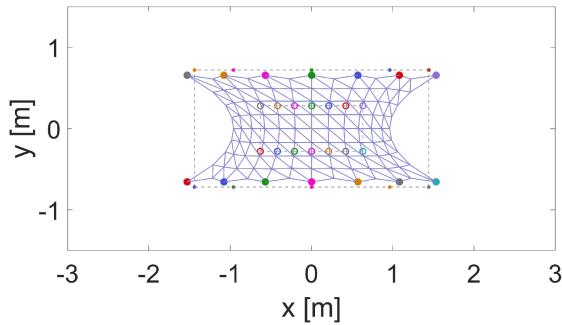
Anexo D

Resultados gráficos en entornos 3D

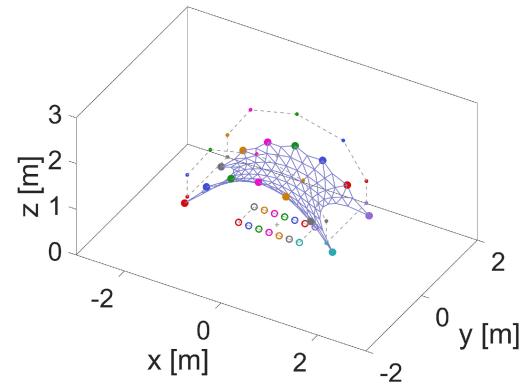
D.1. Gráficas de la simulación 6.1

Este apartado recoge gráficas adicionales de la simulación donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D. Las Figuras D.1 y D.2 muestran una vista completa, además de la vista superior, frontal y lateral de la deformación del objeto en el instante $t = 0.4$ s. Comparando los dos grupos de imágenes se observan las diferencias ya comentadas entre utilizar un control con rotaciones infinitesimales y con el algoritmo de Kabsch.

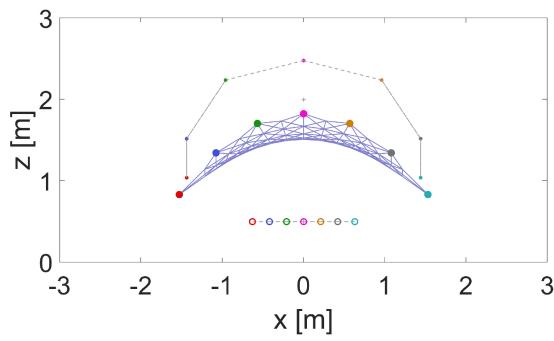
El resto de gráficas hacen referencia a las trayectorias, posiciones y velocidades de los agentes. Nótese que, al no haber rotación y ser la configuración simétrica, muchas de las líneas de estas gráficas aparecen superpuestas sobre otras.



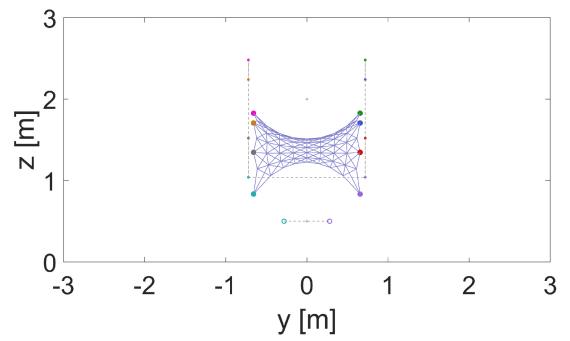
(a) Vista superior



(b) Vista 3D



(c) Vista frontal



(d) Vista lateral

Figura D.1: Deformación de un objeto deformable en el instante $t = 0.4$ s en el experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D al aplicar el control con rotaciones infinitesimales, con $k_G = 5$. Vídeo F.5.

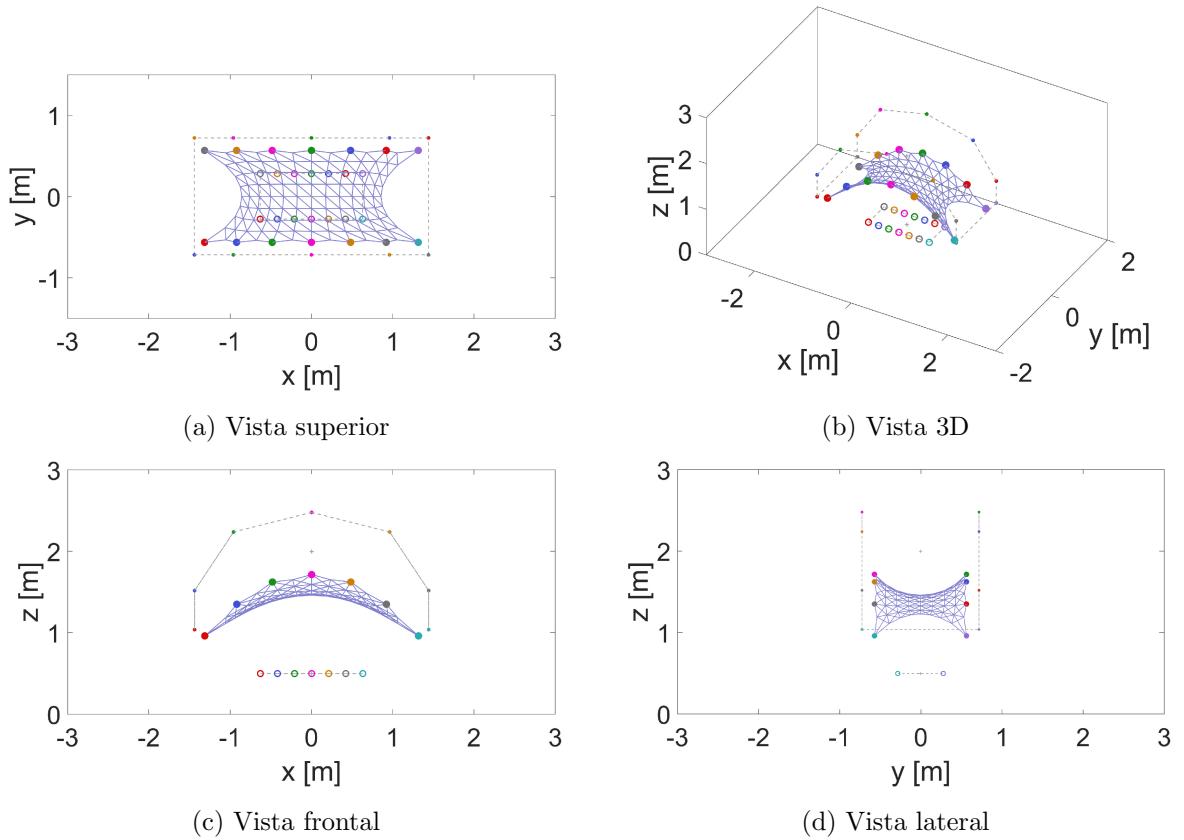


Figura D.2: Deformación de un objeto deformable en el instante $t = 0.4$ s en el experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D al aplicar el control con el algoritmo de Kabsch, con $k_G = 5$. Vídeo F.5.

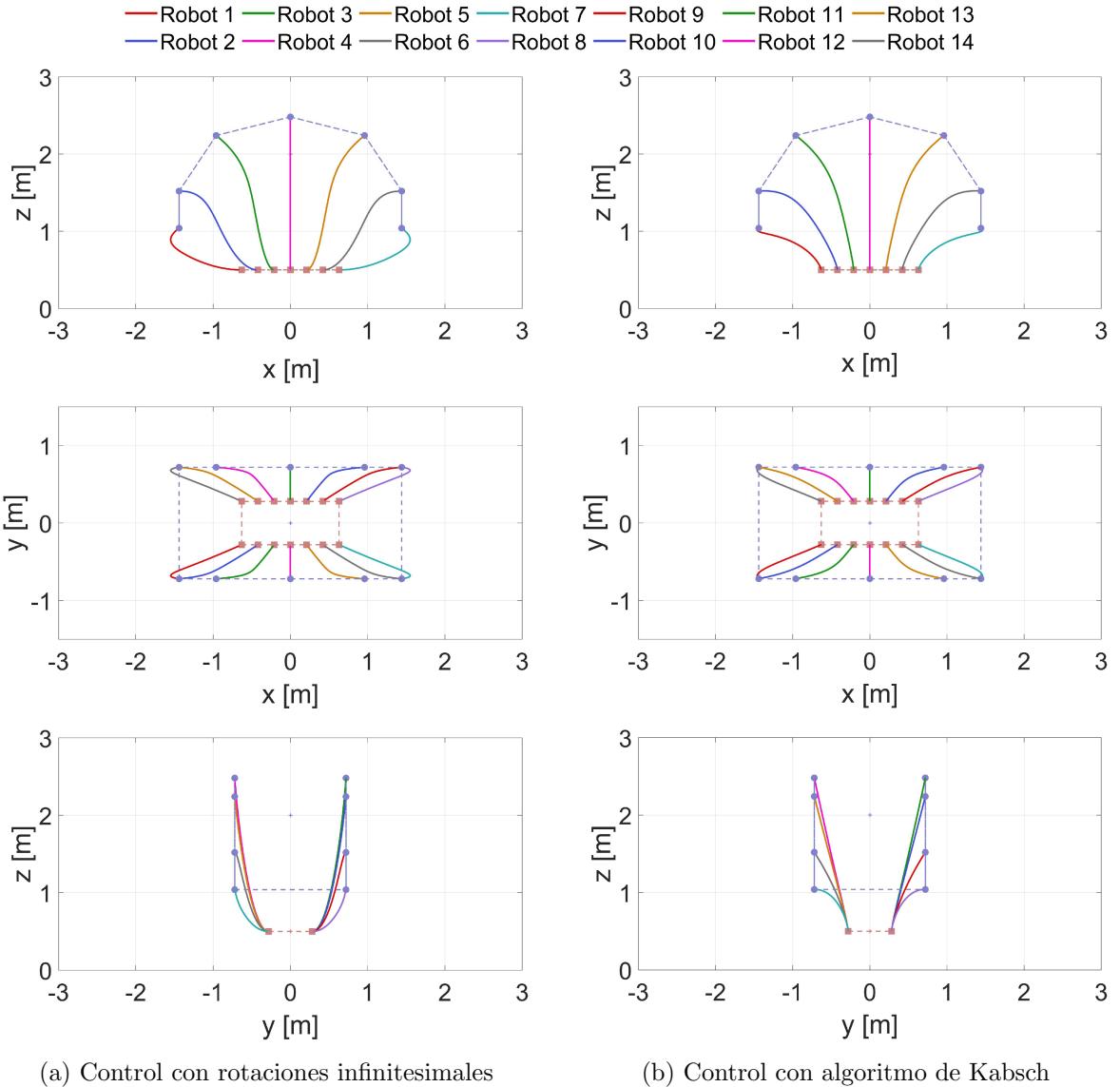
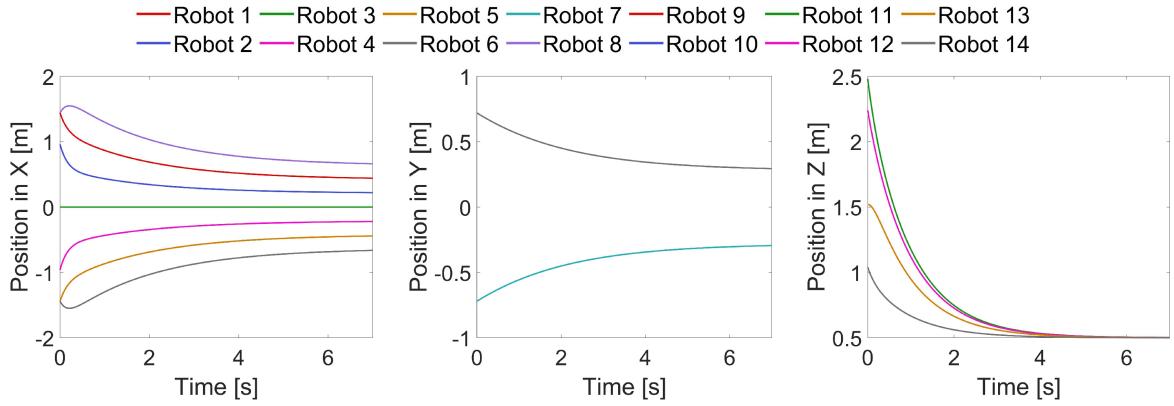
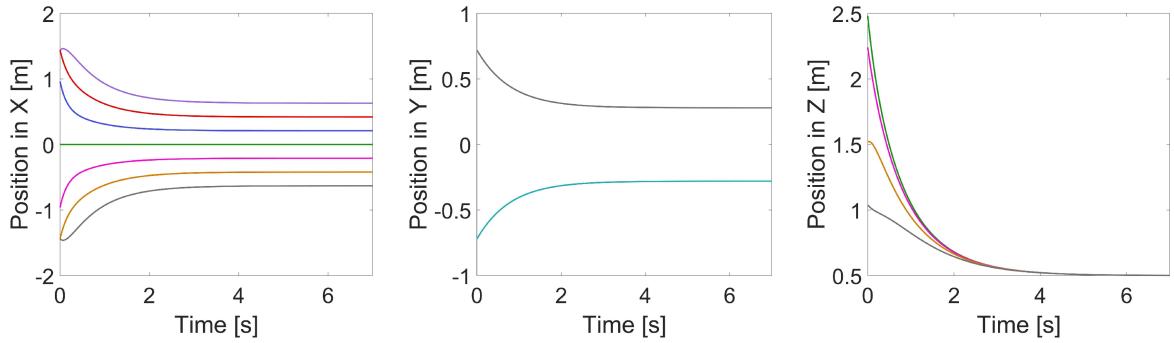


Figura D.3: Comparación de las trayectorias seguidas por los robots al aplicar los dos métodos de control considerados, con $k_G = 5$, del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D. En cada fila se muestra una vista: 1^a, vista frontal; 2^a, vista superior; 3^a, vista lateral.

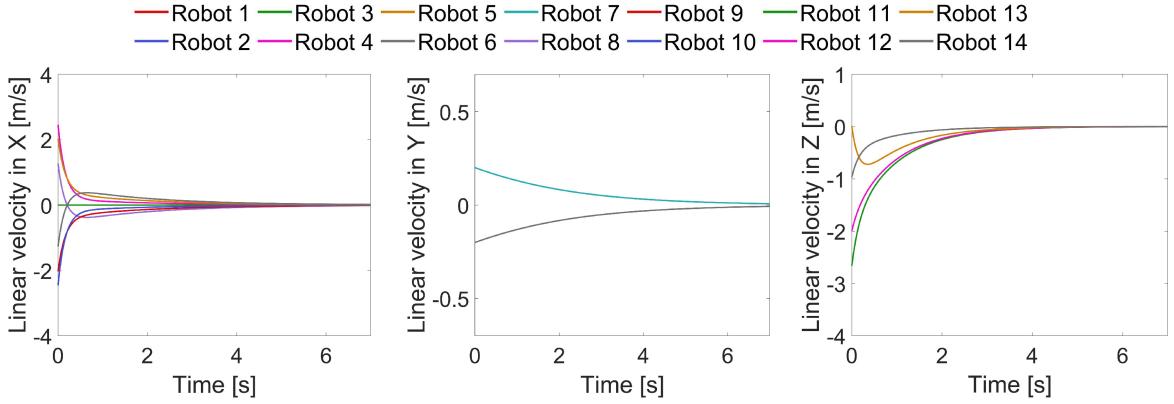


(a) Control con rotaciones infinitesimales

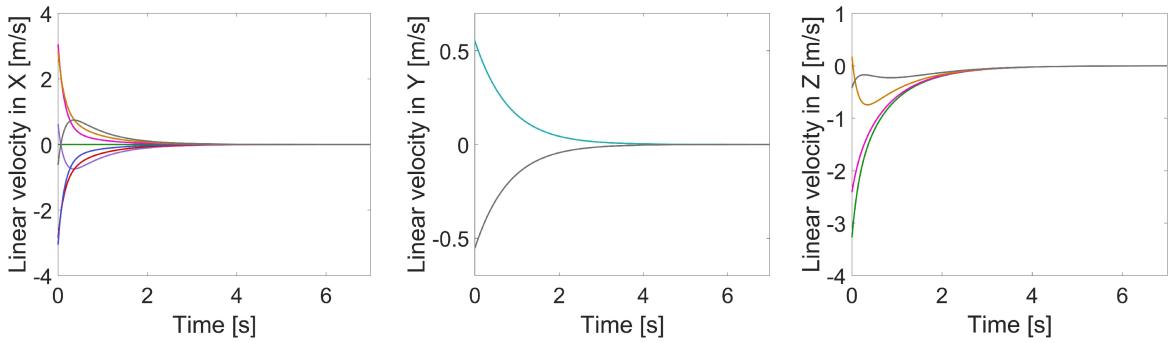


(b) Control con algoritmo de Kabsch

Figura D.4: Comparación de la variación de las coordenadas X, Y y Z de posición al aplicar los dos métodos de control considerados, con $k_G = 5$, del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D.



(a) Control con rotaciones infinitesimales



(b) Control con algoritmo de Kabsch

Figura D.5: Comparación de la variación de las componentes X, Y y Z de velocidad lineal al aplicar los dos métodos de control considerados, con $k_G = 5$, del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D.

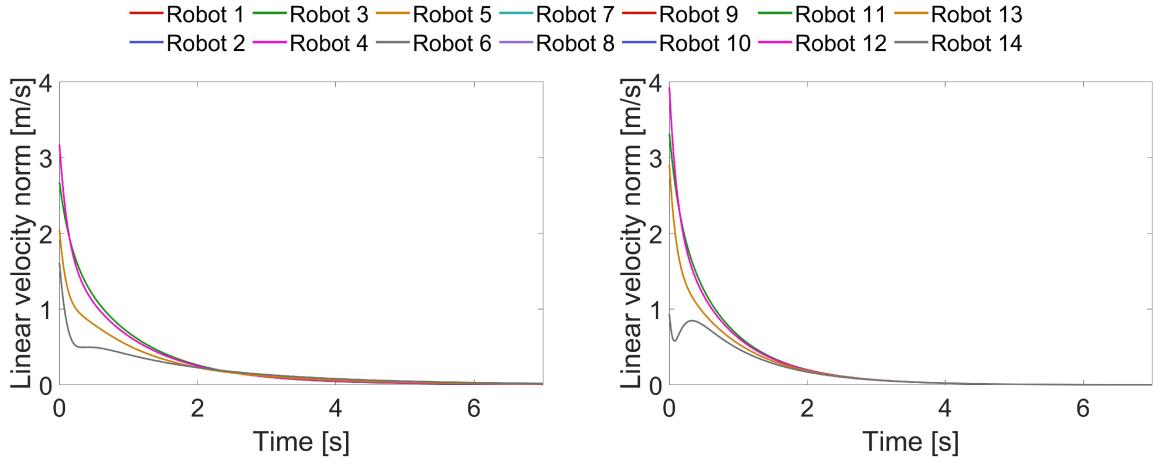


Figura D.6: Comparación de la variación del módulo de velocidad lineal al aplicar los dos métodos de control considerados, con $k_G = 5$, del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D.

D.1.1. Resultados cuando no se controlan las deformaciones

A continuación, se muestran las gráficas de deformación y trayectorias correspondientes al experimento de simulación donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D cuando no se controlan las deformaciones durante el transporte. Es decir, $k_G = 0$. Se observa que los movimientos son más directos que con $k_G = 5$, lo cual genera deformaciones no deseadas en el objeto.

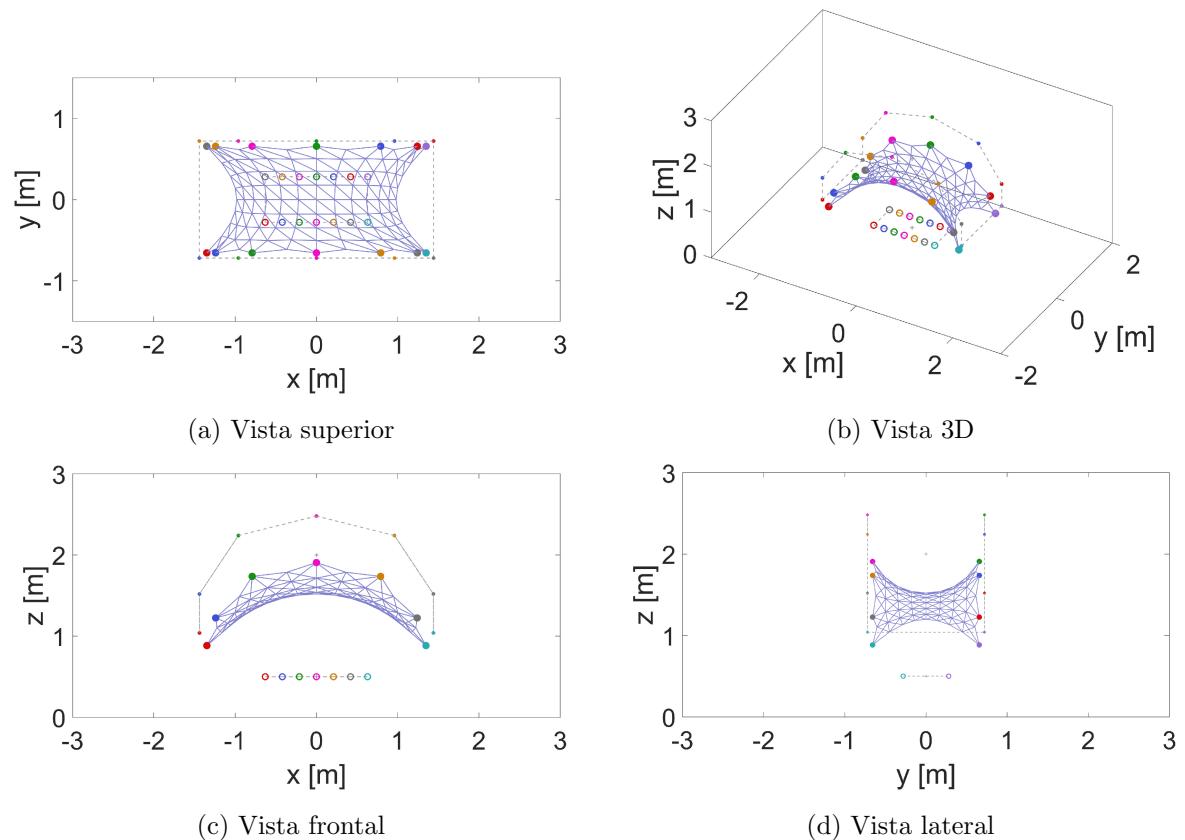
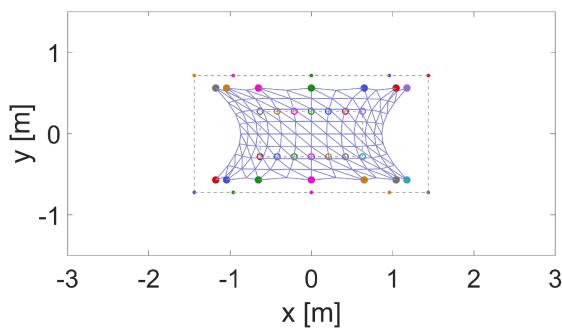
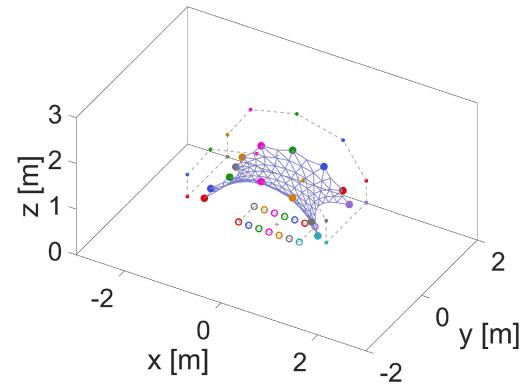


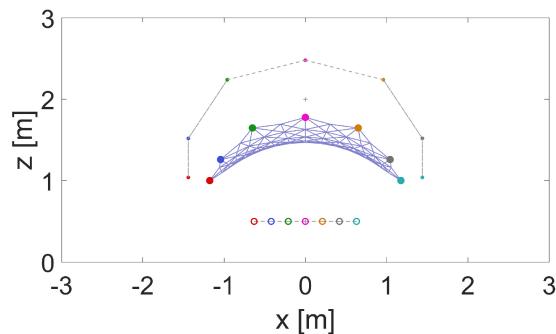
Figura D.7: Deformación de un objeto deformable en el instante $t = 0.4$ s en el experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D al aplicar el control con rotaciones infinitesimales, con $k_G = 0$.



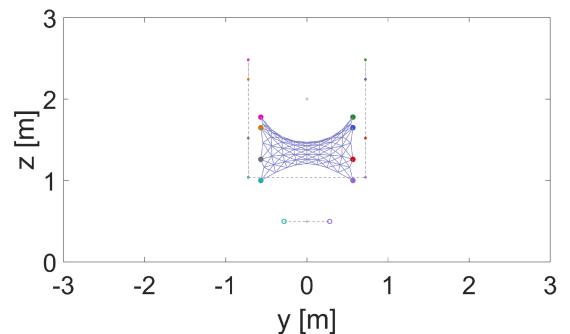
(a) Vista superior



(b) Vista 3D



(c) Vista frontal



(d) Vista lateral

Figura D.8: Deformación de un objeto deformable en el instante $t = 0.4$ s en el experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D al aplicar el control con el algoritmo de Kabsch, con $k_G = 0$.

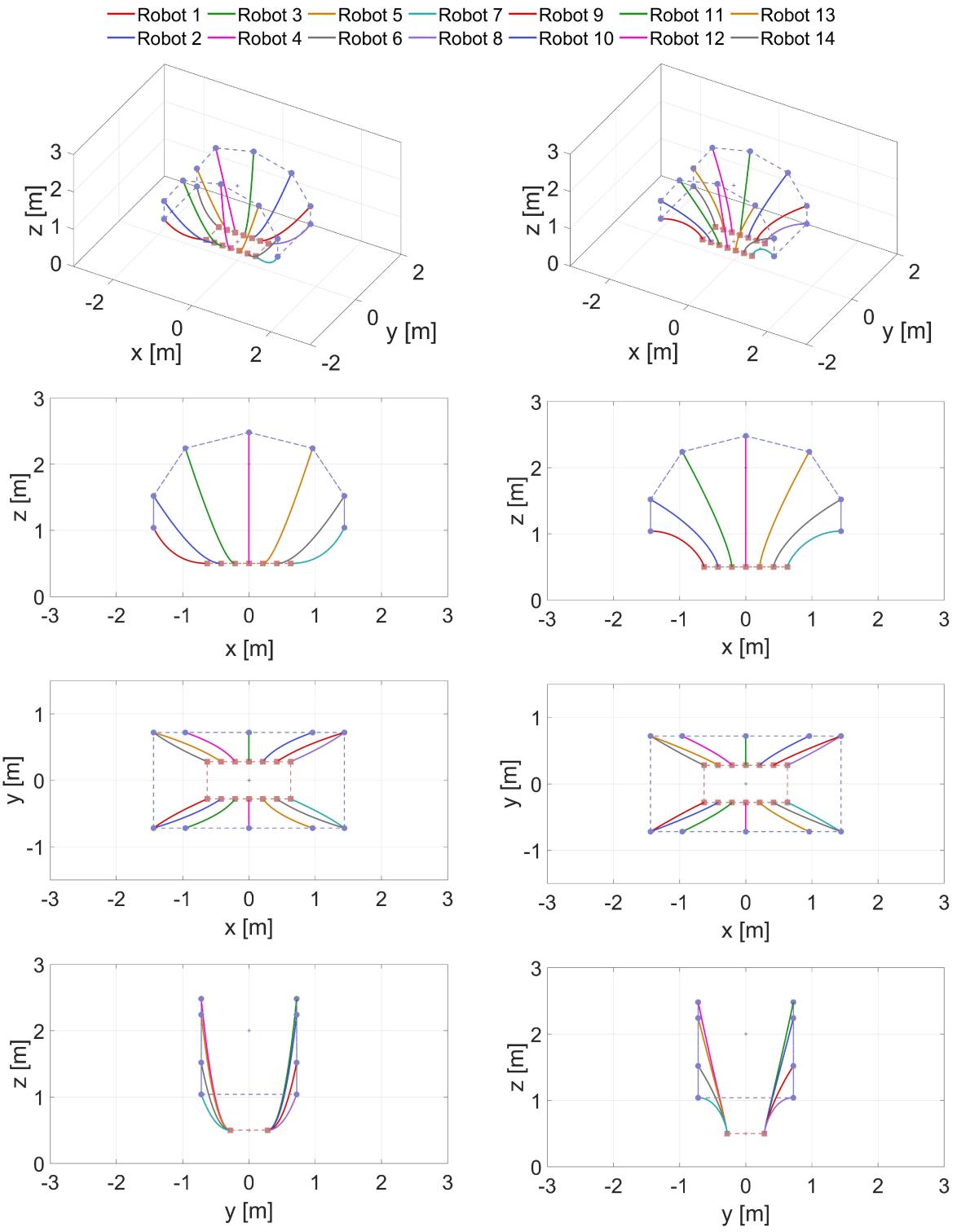


Figura D.9: Comparación de las trayectorias seguidas por los robots al aplicar los dos métodos de control considerados, con $k_G = 0$, del experimento simulado donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular en un espacio 3D. En cada fila se muestra una vista: 1^a, vista 3D; 2^a, vista frontal; 3^a, vista superior; 4^a, vista lateral.

D.2. Gráficas de la simulación 6.2

A continuación, se muestran las gráficas de posiciones y velocidades de la simulación donde el equipo de robots pasa de una configuración deformada a una configuración rectangular y rotada en un espacio 3D. Puesto que en el apartado anterior, así como en los experimentos 2D, ya se han analizado las diferencias entre restringir o no las deformaciones durante el transporte, en esta sección solo se recogen los resultados obtenidos cuando $k_G = 2$.

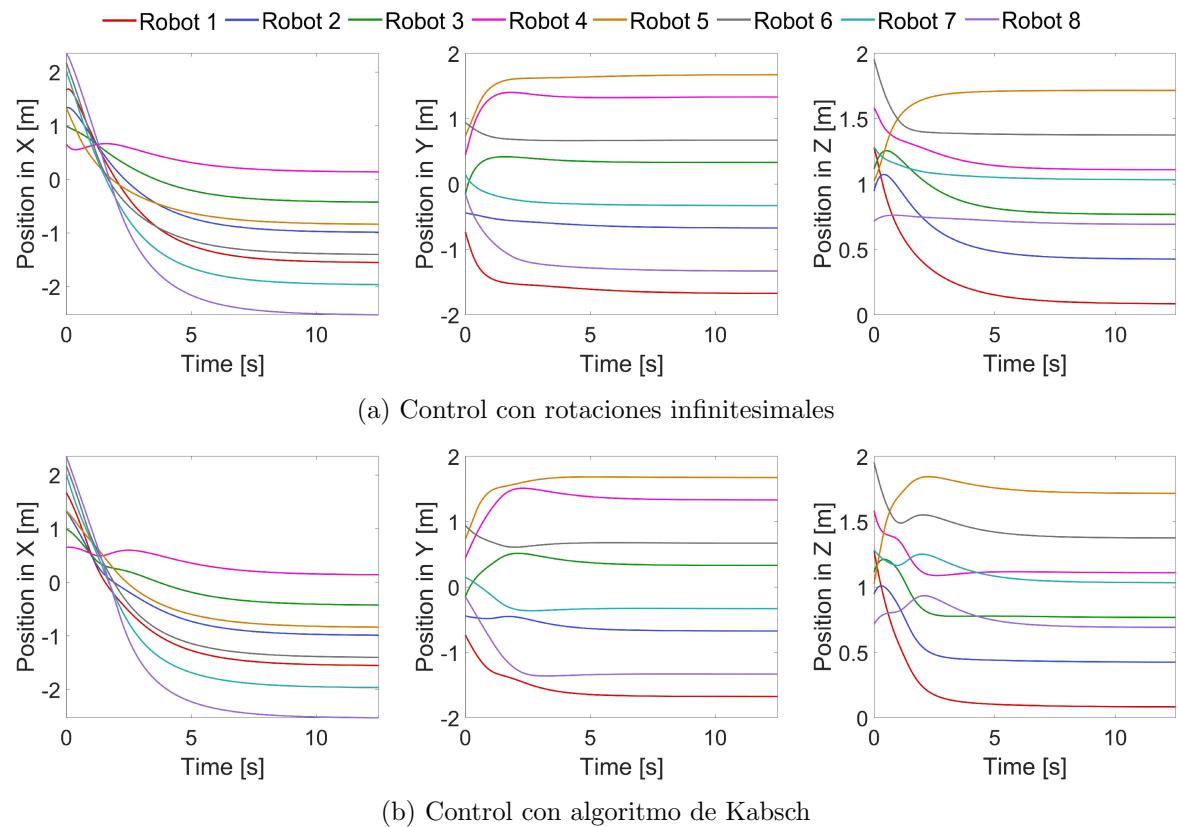
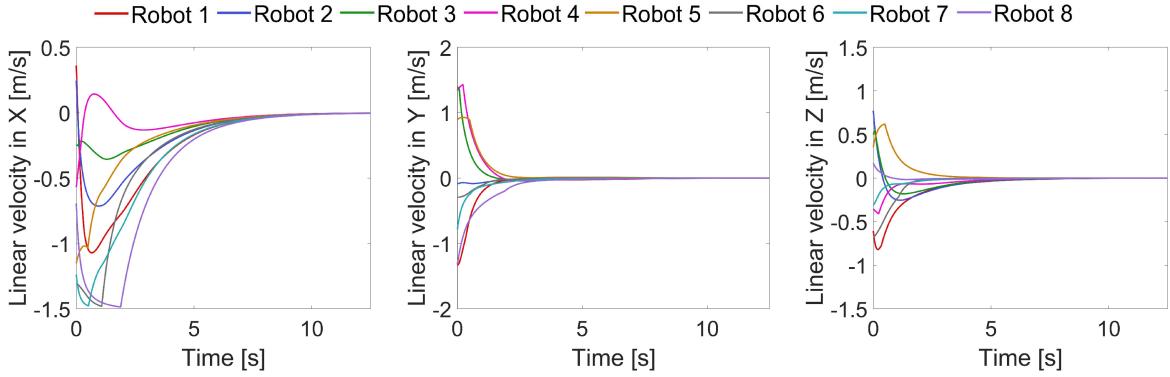
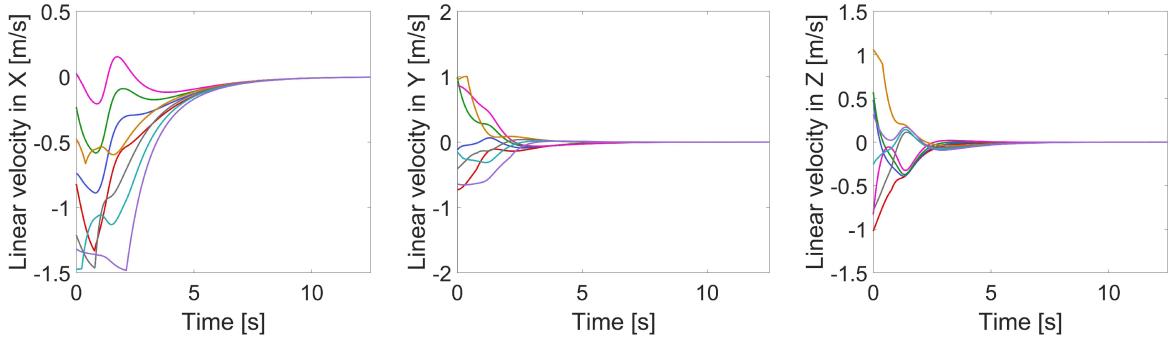


Figura D.10: Comparación de la variación de las coordenadas X, Y y Z de posición al aplicar los dos métodos de control considerados, con $k_G = 2$, del experimento simulado donde el equipo de robots pasa de una configuración deformada a una configuración rectangular y rotada en un espacio 3D.

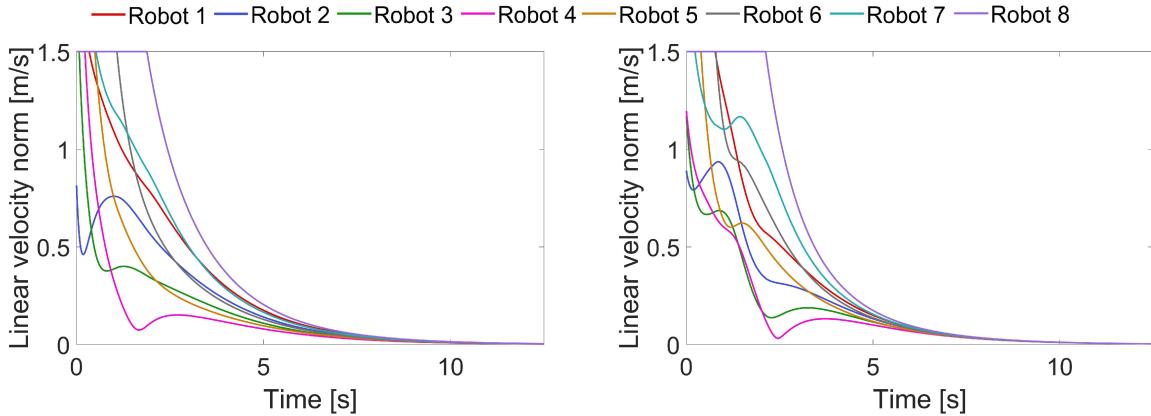


(a) Control con rotaciones infinitesimales



(b) Control con algoritmo de Kabsch

Figura D.11: Comparación de la variación de las componentes X, Y y Z de velocidad lineal al aplicar los dos métodos de control considerados, con $k_G = 2$, del experimento simulado donde el equipo de robots pasa de una configuración deformada a una configuración rectangular y rotada en un espacio 3D.



(a) Control con rotaciones infinitesimales

(b) Control con algoritmo de Kabsch

Figura D.12: Comparación de la variación del módulo de velocidad lineal al aplicar los dos métodos de control considerados, con $k_G = 2$, del experimento simulado donde el equipo de robots pasa de una configuración deformada a una configuración rectangular y rotada en un espacio 3D.

D.2.1. Resultados cuando no se controla la rotación

Un caso particular de este experimento es una situación en la que no se controle la rotación. En el caso de implementar el control con el algoritmo de Kabsch, la formación alcanza la forma, escala y posición deseadas, pero no la orientación. En cambio, al utilizar las rotaciones infinitesimales, los resultados no son los esperados. Considerando las mismas condiciones que en el Apartado 6.2, pero con $k_R = 0$, la formación alcanza un tamaño distinto al de la configuración deseada, tal como refleja la Figura D.13.

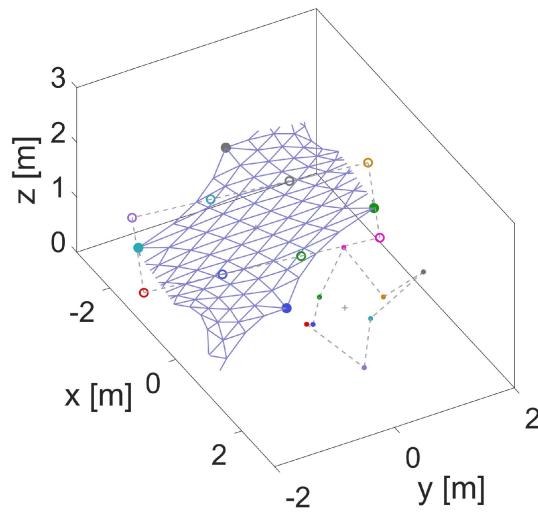


Figura D.13: Representación de la configuración de la formación al aplicar rotaciones infinitesimales, con $k_R = 0$, en el experimento simulado donde el equipo de robots pasa de una configuración deformada a una configuración rectangular y rotada en un espacio 3D. Los puntos de colores son las posiciones iniciales de los robots y los círculos huecos representan las posiciones finales deseadas.

Además, en la Figura D.14, al comienzo de la simulación se observa que, tanto el error de rotación como los errores individuales de orientación en torno a cada eje, se reducen ligeramente. Después, convergen a valores constantes durante el resto del experimento.

Los efectos «no deseados» que se observan en la configuración final se deben a que la matriz \mathbf{R} usada es una aproximación menos precisa que una matriz de rotación verdadera, puesto que difiere bastante de la matriz identidad. En la práctica, estos efectos se pueden evitar utilizando $k_R > 0$ o aplicando el algoritmo de Kabsch antes de que los robots inicien su movimiento para redefinir \mathbf{c} .

Como ya se ha mencionado, esto solo ocurre en este caso concreto. Controlando la rotación, la formación robótica converge correctamente a la configuración deseada.

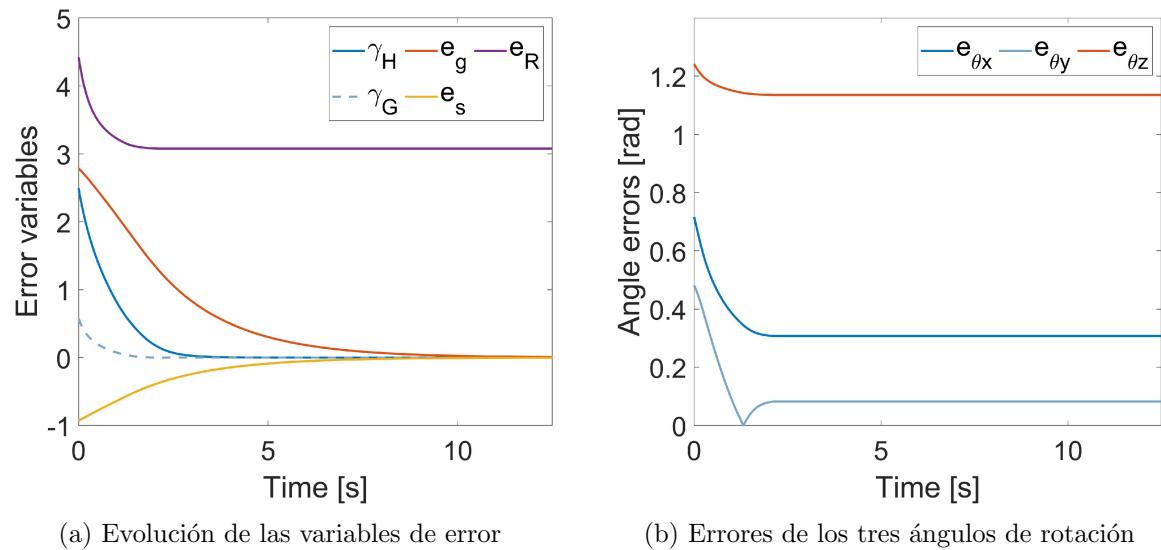


Figura D.14: Representación de la evolución de las variables de error y de los ángulos de rotación al aplicar rotaciones infinitesimales, con $k_R = 0$, en el experimento simulado donde el equipo de robots pasa de una configuración deformada a una configuración rectangular y rotada en un espacio 3D.

D.3. Gráficas de la simulación 6.3

En este apartado se muestran resultados complementarios de la simulación donde el equipo de robots transporta un colchón a lo largo de un pasillo curvo en un espacio 3D. Recordar que el experimento se ha realizado implementando el algoritmo de Kabsch y limitando las deformaciones ($k_G = 5$). En la Figura D.15 se puede ver el movimiento del colchón durante la tarea, manteniendo su forma y escala originales en todo momento hasta llegar a la configuración deseada.

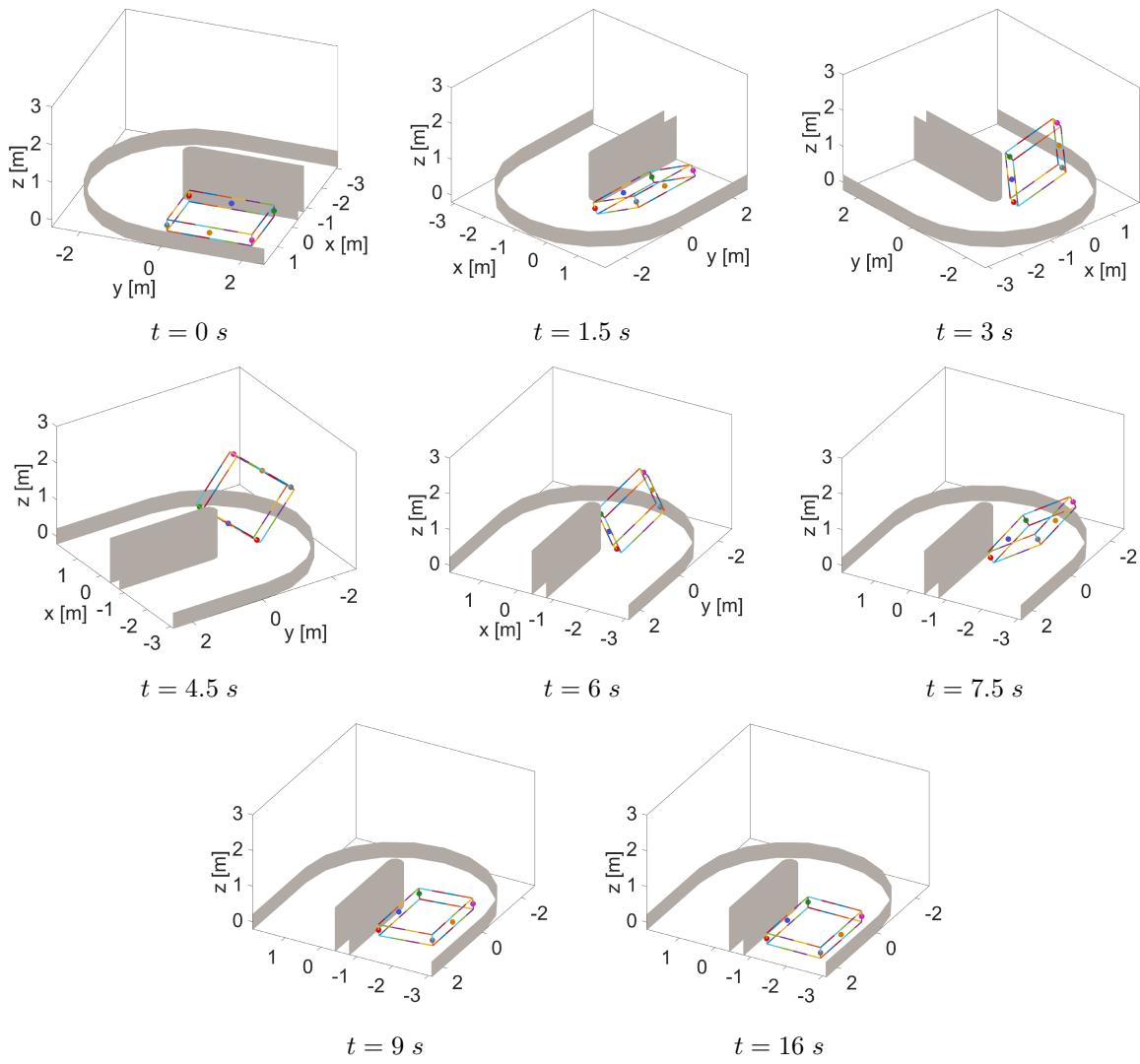


Figura D.15: Deformación de un colchón durante su transporte en simulación a lo largo de un pasillo curvo en un espacio 3D.

Las siguientes gráficas reflejan la evolución de las posiciones y velocidades de cada robot durante la tarea de transporte.

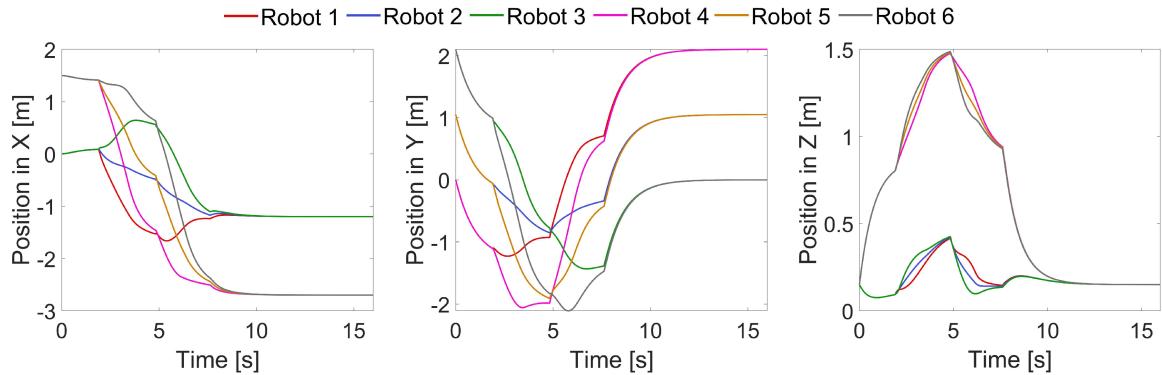


Figura D.16: Representación de la variación de las coordenadas X, Y y Z de posición al aplicar el algoritmo de Kabsch, con $k_G = 5$, del experimento simulado donde el equipo de robots transporta un colchón a lo largo de un pasillo curvo en un espacio 3D.

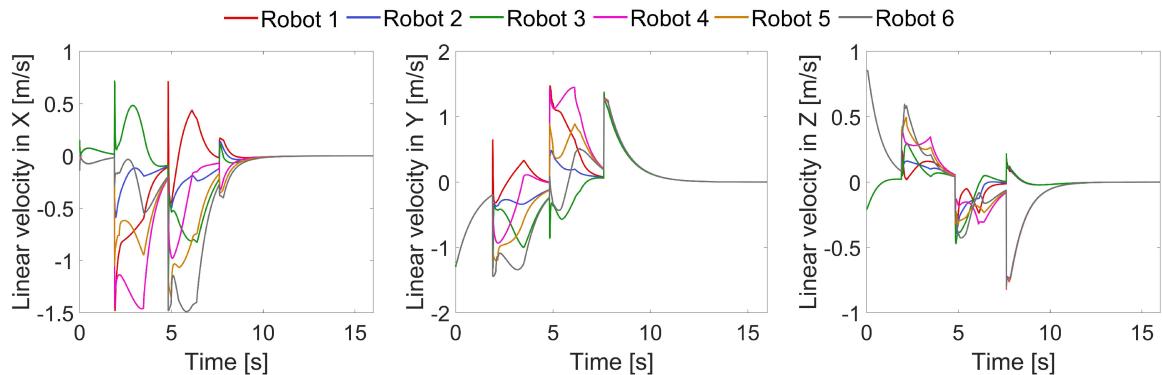


Figura D.17: Representación de la variación de las componentes X, Y y Z de velocidad lineal al aplicar el algoritmo de Kabsch, con $k_G = 5$, del experimento simulado donde el equipo de robots transporta un colchón a lo largo de un pasillo curvo en un espacio 3D.

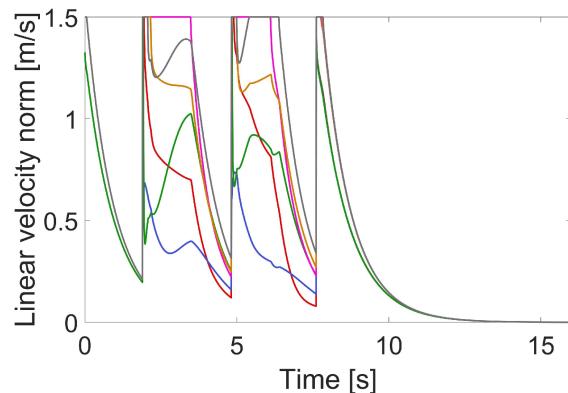


Figura D.18: Representación de la variación del módulo de velocidad lineal al aplicar el algoritmo de Kabsch, con $k_G = 5$, del experimento simulado donde el equipo de robots transporta un colchón a lo largo de un pasillo curvo en un espacio 3D.

Anexo E

Fragmentos de código

En este Anexo se recoge el *script* principal del algoritmo de control implementado en los experimentos 2D. Por simplicidad, la definición de los diferentes parámetros de cada configuración, así como de la forma de la configuración de referencia, se han definido en archivos independientes (*create_exp_parameters* y *create_reference_conf*, respectivamente). Lo mismo ocurre con el diseño de los obstáculos (*create_obstacles*) y la definición de los *waypoints* (*create_waypoints*).

El controlador *multirobot_controller* se describe también en un archivo individual. En él, se ha programado el conjunto de ecuaciones de control definidas en el Capítulo 3. Puesto que los ensayos en entornos 2D se han realizado con Robotarium, en el código se incluye la implementación en dicha plataforma.

La metodología para ejecutar los experimentos en entornos 3D es similar. Basta con sustituir las ecuaciones de control por las indicadas en el Capítulo 5 y eliminar del código aquellas partes que involucren a Robotarium.

```
1 % Multirobot control in a 2D workspace with a team of unicycle mobile robots for
2 % deformable object transport
3
4 % The goal of the controller is to keep, to the extent possible, the shape
5 % of the robots formation on its way from an initial to a desired configuration
6 % with different shape, size, centroid and rotation while allowing linear
7 % and quadratic deformations.
8
9 % Author: Raquel Marcos Saavedra, February 2024
10
11 %%%%%%%%%%%%%%
12 %%%%%%%%%%%%%%
13
14 ob_type = 1; % 1 - for sheet object
15 with_obstacles = 0; % 1 - if the experiment has obstacles
16 with_waypoints = 0; % 1 - if the experiment has waypoints
17
18 %% We define the object's nodes arranged in a rectangular grid
```

```

19
20 nx = 7;      % number of nodes in x direction of the grid
21 ny = 7;      % number of nodes in y direction of the grid
22
23 sx = 2;      % length of the object in x direction (in the graphics, it will
24 % be the 'y' direction)
25 sy = 3;      % length of the object in y direction (in the graphics, it will
26 % be the x direction)
27
28 % We need to reduce the defined object for the Robotarium experiment because
29 % its working area is smaller.
30 scale_ob = 0.3;
31
32 % mesh of the object: its shape and the links between nodes
33 omesh = create_mesh(nx, ny, sx * scale_ob, sy * scale_ob);
34 Pob0 = omesh.shape'; % "rest" shape of the object (x,y,z coordinates for each node)
35
36
37 %% We initialize the model (As-Rigid-As-Possible) of the deformable object
38 % And we also define the desired configuration of the team of robots, as they are
39 % attached to the object
40
41 % The sheet is simulated in 3D, but we will only display its projection
42 % on the XY plane
43
44 handled_nodes = [1 15 29 43 49 35 21 7]; % the nodes of the object where we place
45 % the robots
46 N = length(handled_nodes); % number of robots (equal to the number
47 % of object nodes they are attached to)
48 K = kron(eye(N)-(1/N)*ones(N,N), eye(2)); % centering matrix 2Nx2N
49 S = [0 -1; 1 0];
50 T = kron(eye(N),S);
51
52 % We define the parameters of the initial and desired configurations
53 % (centroids, scales and orientations) for each experiment in order to
54 % make it easier to select them.
55 shape = 'default'; % shape of the reference configuration we want in the
56 % current experiment
57 create_exp_parameters; % execute the script with the parameters definition
58
59 % We define the initial configuration for the team of robots, p0
60
61 % Positions of the robots (2D) with centroid (0,0), before rotating - vector 2Nx1
62 p00 = K * reshape(Pob0(1:2, handled_nodes), [2*N,1]);
63
64 R00 = [cos(th0) -sin(th0); sin(th0) cos(th0)]; % rotation matrix 2x2
65 R0 = kron(eye(N), R00); % rotation matrix 2Nx2N
66
67 % Initial positions of the robots (2D) with centroid g0 - vector 2Nx1
68 p0 = (s0 * R0 * p00) + reshape(g0 * ones(1,N), [2*N,1]);
69
70 % Value of 2 or 3: the higher the value the stiffer the simulated object
71 n_iters_arap = 3;
72 % Create model parameters
73 arap_params = create_params_for_arap(omesh, handled_nodes, [], n_iters_arap);
74
75 % We define the reference configuration, c
76 create_reference_conf; % execute the script with the reference configuration
77 % definition
78
79 % We define the target configuration for the team of robots, pd
80 pd0 = c; % the target shape is the same as the reference one (but we will
81 % scale it, rotate it, translate it), centroid (0,0)
82 hd0 = [sd0*cos(thd0), sd0*sin(thd0)]; % desired shape preserving
83 % transformation
84 Rd00 = [cos(thd0) -sin(thd0); sin(thd0) cos(thd0)]; % rotation matrix 2x2
85 Rd0 = kron(eye(N), Rd00); % rotation matrix 2Nx2N
86 pd = (sd0 * Rd0 * pd0); % final rotated and scaled configuration with centroid (0,0)
87
88 % Target formation scale and rotation. Since we have already rotated and scaled
89 % pd when we defined it above, then we can select sd = 1 and thd = 0.
90 sd = 1; % [radians]
91 thd = 0;

```

```

92 Rd = [cos(thd) -sin(thd); sin(thd) cos(thd)]; % rotation matrix 2x2
93 Rd = kron(eye(N), Rd); % rotation matrix 2Nx2N
94 % Target configuration
95 PT = (sd * Rd * pd) + reshape(gd * ones(1,N), [2*N,1]); % final configuration ,
96 % centroid in gd
97
98 % Control terms are calculated with c. As the code is defined here, we can
99 % replace c with pd in the control terms code or do the following:
100 c = K * PT;
101
102
103 %% Waypoints definition
104
105 num_waypoint = 1;
106 if with_waypoints == 1
107     waypoint_design = 'default'; % name of the waypoints design defined
108     % in create waypoints
109     [waypoints_gd, waypoints_c] = create waypoints(ob_type, waypoint_design, ...
110     gd, c, N, p00, K);
111     waypoint_gd = waypoints_gd(:,1); % centroid of the first waypoint
112     waypoint_c = waypoints_c(:,1); % configuration of the first waypoint
113 else
114     waypoints_gd = gd;
115     waypoints_c = c;
116     waypoint_gd = waypoints_gd;
117     waypoint_c = waypoints_c;
118 end
119
120
121 %% Initial conditions of the simulation and required simulation parameters
122
123 % Initial configuration of the object
124 Pob = Pob0;
125
126 % Initial positions of the robots
127 p = p0; % p matrix 2Nx1
128
129 % Initial velocity of the robots
130 v = 0;
131
132 % Time parameters - The simulated time in seconds will be nit*dt
133 dt = 0.033; % time step of the simulation
134 niters = 4500; % number of iterations of the control loop
135
136 %%%%%%%%%%%%%%
137
138 % Gain values for the control
139 k_H = 0.2; % for moving toward shape-preserving transformation
140 k_G = 1; % for moving toward a configuration consistent with our deformation modes
141 k_c = 0.1; % for centroid translation
142 k_s = 0.1; % for scaling
143 k_th = 0.1; % for rotation
144
145 %%%%%%%%%%%%%%
146
147 % Each of the following will be a growing array that will store the values of variables
148 ps = []; % positions
149 gammas_H = []; % cost relative to shape-preserving transformation
150 gammas_G = []; % cost relative to configuration consistent with our deformation modes
151 hs = []; % parameters of the optimal shape-preserving transformation
152 egs = []; % centroid errors
153 ess = []; % scale errors
154 eths = []; % angle errors
155 dx_unis = []; % unicycle velocities
156 vs = []; % linear velocities
157 nvs = []; % linear velocities norm
158
159
160 %% Obstacles definition
161 obstacles = [];
162 obstacle_design = 'default'; % name of the obstacles design defined in create obstacles
163
164 if with obstacles == 1

```

```

165     create_obstacles;
166 end
167
168
169 %% Robotarium Initialization
170 r = Robotarium('NumberOfRobots', N, 'ShowFigure', true, 'InitialConditions',...
171     [reshape(p0,[2,N]); thr0]);
172
173 %% Projection parameters
174 projection_distance = 0.05;
175 projected_gd = gd;
176 angle_c = zeros(1,N);
177 projected_c = zeros(2*N,1);
178 project_gd = 1;
179 data_current.g = g0;
180
181 %% Security distance (margin), robots will stop if the distance between
182 %% them and other elements is < margin
183 margin = 0.02;
184
185 %% Angular velocity limit
186 ang_vel_limit = pi/10;
187
188 %%%%%%%%%%%%%%
189
190 %% We initialize controllers and barrier function to avoid collisions
191 multirobot_controller = create_multirobot_controller_def_modes(...%
192     'ref_shape', waypoint_c, 'gd', waypoint_gd, 'sd', sd, ...%
193     'thd', thd, 'hd', hd0, 'k_H', k_H, 'k_G', k_G, ...%
194     'k_c', k_c, 'k_s', k_s, 'k_th', k_th, 'k_Hd', k_Hd);
195
196 [si_to_uni_dynamics, uni_to_si_states] = create_si_to_uni_mapping(...%
197     'ProjectionDistance', projection_distance);
198 barrier_certificate = create_uni_barrier_certificate_with_boundary();
199
200 %% We also need to initialize the controller using the final reference configuration
201 %% in order to calculate the real error between initial and target configurations.
202 calculate_current_parameters = create_multirobot_controller_def_modes(...%
203     'ref_shape', c, 'gd', gd, 'sd', sd, 'thd', thd, ...%
204     'hd', hd0, 'k_H', k_H, 'k_G', k_G, 'k_c', k_c, ...%
205     'k_s', k_s, 'k_th', k_th, 'k_Hd', k_Hd);
206
207 %% We save the current initial positions of the robots
208 robot_current_pos = r.get_poses(); % matrix 3xN (x,y,thr of each robot)
209 r.step();
210
211
212 %% Control loop
213 for it_loop = 1:niters
214
215     robot_current_pos = r.get_poses(); % we save the current states (position +
216     % orientation) of the robots
217
218     %% We update the desired centroid for the controller
219     if num_waypoint < size(waypoints_gd,2)
220         if norm(data_current.g - waypoint_gd) < 0.3
221             num_waypoint = num_waypoint + 1;
222             waypoint_gd = waypoints_gd(:, num_waypoint);
223             waypoint_c = waypoints_c(:, num_waypoint);
224             multirobot_controller = create_multirobot_controller_def_modes(...%
225                 'ref_shape', waypoint_c, 'gd', waypoint_gd, 'sd', sd, ...%
226                 'thd', thd, 'hd', hd0, ...%
227                 'k_H', k_H, 'k_G', k_G, 'k_c', k_c, 'k_s', k_s, ...%
228                 'k_th', k_th, 'k_Hd', k_Hd);
229     end
230
231     else % We project the target centroid to correct the final position error
232         distance_g = data_current.g - waypoint_gd;
233         if norm(distance_g) > 0.02
234             angle_g = atan2(distance_g(2), distance_g(1));
235             projected_gd(1) = waypoint_gd(1) - (projection_distance * cos(angle_g));
236             projected_gd(2) = waypoint_gd(2) - (projection_distance * sin(angle_g));
237

```

```

238     multirobot_controller = create_multirobot_controller_def_modes(...  

239                     'ref_shape', waypoint_c, 'gd', projected_gd, 'sd', sd, ...  

240                     'thd', thd, 'hd', hd0, ...  

241                     'k_H', k_H, 'k_G', k_G, 'k_c', k_c, 'k_s', k_s, ...  

242                     'k_th', k_th, 'k_Hd', k_Hd);  

243     end  

244  

245 end  

246  

247 %%%%%%%%%%%%%%  

248  

249 % MULTIROBOT CONTROL  

250  

251 si_states = uni_to_si_states(robot_current_pos); % matrix 2xN - Projects the  

252 % single-integrator system a distance  

253 % in front of the unicycle one  

254 si_states = reshape(si_states, [2*N, 1]); % matrix 2Nx1  

255  

256 [v, data] = multirobot_controller(si_states); % v - matrix 2Nx1  

257 v = reshape(v, [2, N]); % v - matrix 2xN  

258  

259 % We execute the controller again just to calculate the error variables with  

260 % respect to the target configuration and with the actual robot positions,  

261 % not the projectated ones. Velocity values are useless here.  

262 [v_useless, data_current] = calculate_current_parameters(...  

263             reshape(robot_current_pos(1:2, :), [2*N, 1]));  

264  

265 % To stop each robot when it is near its final position  

266 distance_to_target = reshape(PT, [2, N]) - robot_current_pos(1:2, :);  

267  

268 for i = 1:N  

269     if (norm(distance_to_target(1:2, i)) <= margin) & (norm(distance_g) < 0.005)  

270         v(:, i) = zeros(2, 1);  

271     end  

272 end  

273  

274  

275 % We compute the norm of the velocity  

276 for i = 1:N  

277     nv(i) = norm(v(:, i)); % we store the norms of velocity  

278 end  

279  

280 dx_uni = si_to_uni_dynamics(v, robot_current_pos); % we adapt the velocity to one  

281 % suitable for unicycle dinamics  

282 dx_uni(2, :) = ang_vel_limit * dx_uni(2, :) / (pi/2);  

283  

284 % We apply the barrier function to avoid collisions  

285 dx_uni = barrier_certificate(dx_uni, robot_current_pos, obstacles);  

286  

287 r.set_velocities(1:N, dx_uni);  

288 r.step();  

289  

290 %%%%%%%%%%%%%%  

291  

292 % We store current positions  

293 ps = [ps; robot_current_pos(1:2, :)];  

294  

295 % We store current control errors  

296 gammas_H = [gammas_H; data_current.gamma_H];  

297 gammas_G = [gammas_G; data_current.gamma_G];  

298 egs = [egs; norm(data_current.g - gd)];  

299 ess = [ess; data_current.s - sd];  

300 eths = [eths; abs(data_current.th - thd)];  

301 dx_unis = [dx_unis; dx_uni];  

302 vs = [vs; v];  

303 nvs = [nvs; nv];  

304  

305 end  

306  

307 r.debug();  

308

```


Anexo F

Lista de vídeos

Los vídeos de todos los experimentos mostrados en este trabajo se han publicado en YouTube, cuyos enlaces se muestran a continuación. Se puede acceder directamente a la lista de reproducción completa en: https://youtube.com/playlist?list=PLjm-KzNA87Cr5kKu05J4Xq5I8yQy9On7E&si=iK_4s6U0UcPbFV1q

F.1. Vídeo de transporte en entorno 2D sin obstáculos

En este vídeo se comparan los dos casos considerados en el Apartado 4.2, donde el equipo de robots pasa de una configuración puramente doblada a una configuración rectangular. Se aprecian las diferentes deformaciones que sufre el objeto en cada caso. Enlace: <https://youtu.be/TOWJh2FIV6k>

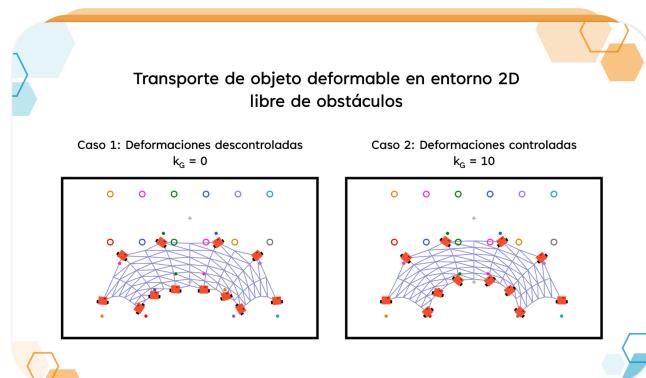


Figura F.1: Vídeo de transporte de objeto deformable en entorno 2D libre de obstáculos.

F.2. Vídeo de transporte en entorno 2D con obstáculo

Se comparan los dos casos considerados en el Apartado 4.3, donde el equipo de robots debe esquivar un obstáculo circular para llegar hasta la configuración final deseada. Enlace: <https://youtu.be/WBm3PmGxnnc>

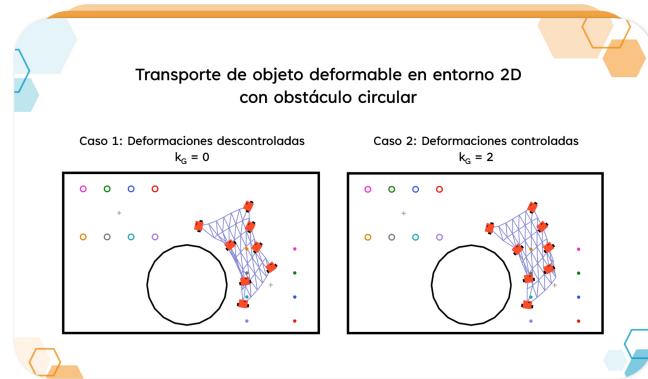


Figura F.2: Vídeo de transporte de objeto deformable en entorno 2D con obstáculo circular.

F.2.1. Vídeo de conservación de forma durante el transporte

Se muestra el efecto en el transporte de un objeto deformable cuando se quiere mantener su forma original durante toda la tarea, tal como se ha comentado en el Anexo C.2.1. Enlace: https://youtu.be/_yszU12wJSw

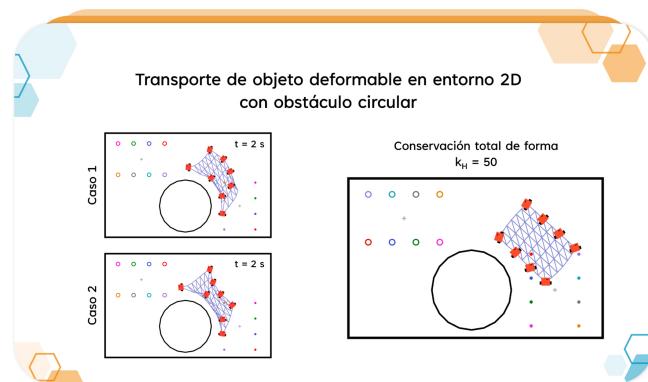


Figura F.3: Vídeo de conservación de forma durante el transporte de objeto deformable en entorno 2D con obstáculo circular.

F.3. Vídeo de transporte en entorno 2D por pasillo curvo

Se comparan los tres casos considerados en el Apartado 4.4, donde el equipo de robots recorre un pasillo curvo en un entorno 2D.

F.3.1. Vídeo de simulación

Simulaciones de los tres casos. Enlace: <https://youtu.be/uKMtPEytEL4>

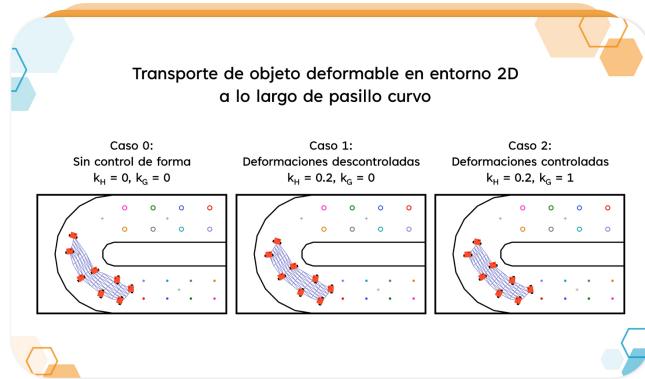


Figura F.4: Vídeo de simulación de transporte de objeto deformable en entorno 2D por pasillo curvo.

F.3.2. Vídeo con robots reales

Ensayos reales en Robotarium de los tres casos. Enlace: <https://youtu.be/dE9uZ2wGbb4>

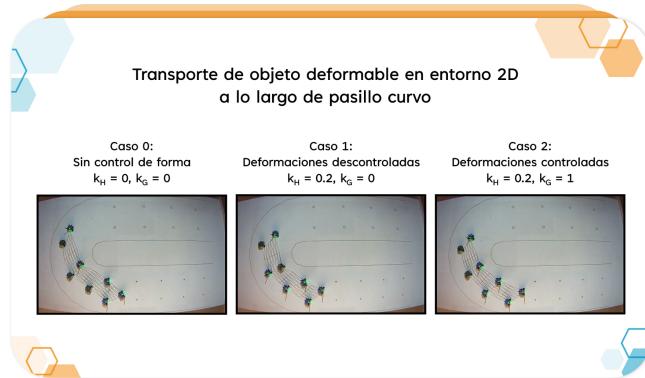


Figura F.5: Vídeo de experimento real de transporte de objeto deformable en entorno 2D por pasillo curvo.

F.4. Vídeo de transporte en entorno 2D con diferente número de robots

Se comparan las tres situaciones del experimento del Anexo C.4, donde se transporte un objeto deformable desde una configuración deformada hasta una configuración rectangular en un entorno 2D. Enlace: <https://youtu.be/1w7Od9VmXeU>

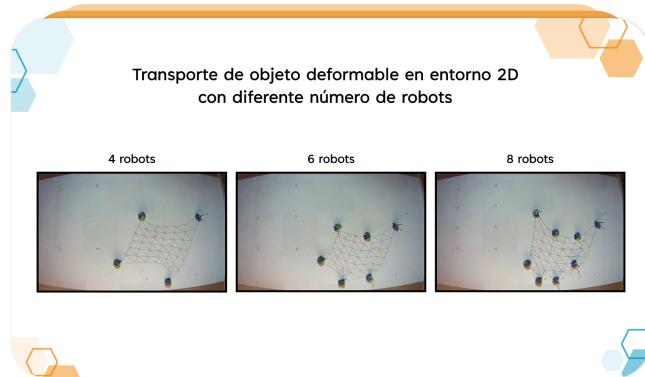


Figura F.6: Vídeo de experimento de transporte de objeto deformable en entorno 2D con diferente número de robots.

F.5. Vídeo de transporte en entorno 3D sin rotación

Se muestra la comparación en la deformación del objeto del experimento del Apartado 6.1 al aplicar un control con rotaciones infinitesimales y un control con el algoritmo de Kabsch. Enlace: <https://youtu.be/44VsutZjQeE>

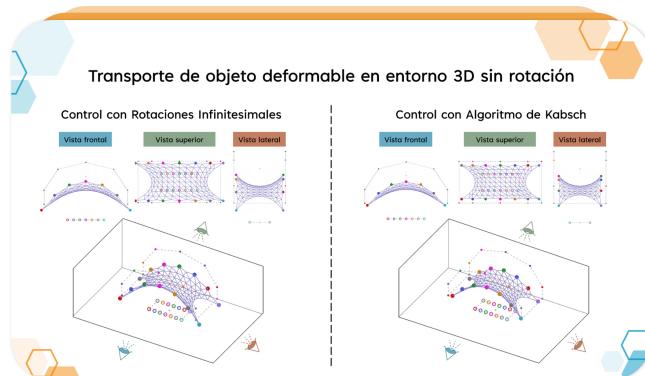


Figura F.7: Vídeo de experimento de transporte de objeto deformable en entorno 3D sin rotación.

F.6. Vídeo de transporte en entorno 3D con rotación

Se compara el movimiento realizado por los robots durante el transporte de un objeto deformable en el experimento del Apartado 6.2. Enlace: <https://youtu.be/gvfp5jWNhml>

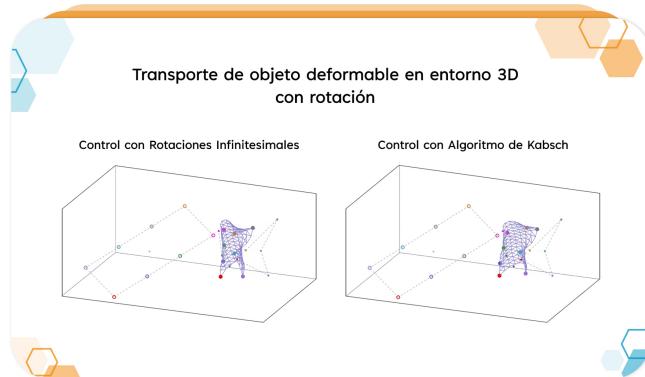


Figura F.8: Vídeo de experimento de transporte de objeto deformable en entorno 3D con rotación.

F.7. Vídeo de transporte de objeto deformable volumétrico

Se muestra la deformación que sufre un colchón en su transporte en el experimento del Apartado 6.3. Enlace: <https://youtu.be/HCGDKxh5TnQ>

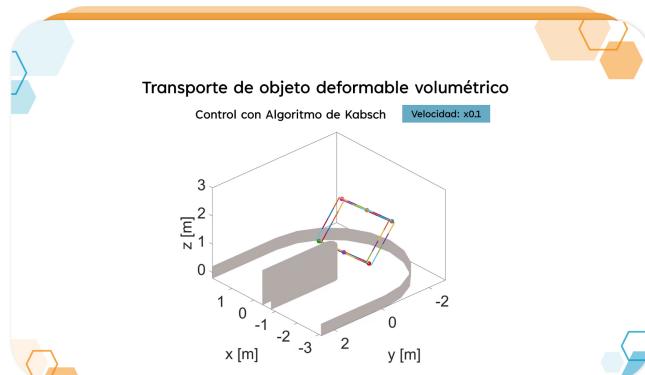


Figura F.9: Vídeo de experimento de transporte de objeto deformable volumétrico.