

**UNIVERSIDAD COMPLUTENSE DE MADRID**  
**FACULTAD DE CIENCIAS FÍSICAS**

DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y AUTOMÁTICA



**TRABAJO DE FIN DE GRADO**

Código de TFG: ACA07

Control de sistemas multiagente y sus aplicaciones

Control of multi-agent systems and their applications

Supervisor/es: Lía García Pérez y Juan Jiménez Castellanos

**YanRu Wu Jin**

Grado en Física

Curso académico 2023-2024

Convocatoria Extraordinaria

# **Control de sistemas multiagente y sus aplicaciones**

## **Resumen:**

En este trabajo se aborda el control de formaciones afines en sistemas multiagente mediante algoritmos de control distribuido. A través de simulaciones y análisis, demostramos la eficacia de estos algoritmos para gestionar y coordinar formaciones complejas. Se introducen los fundamentos matemáticos de la teoría de grafos para control de sistemas multiagente y se realizarán simulaciones para estudiar el comportamiento del modelo construido. Por último se examinarán algunas de las posibles aplicaciones de este tipo de control y se comentarán algunos de los problemas abiertos en el sector, junto con algunas de las posibles mejoras implementables en el modelo.

## **Abstract:**

In this project we address the control of affine formations in multi-agent systems through distributed control algorithms. Through simulations and analysis, we demonstrate the effectiveness of these algorithms in managing and coordinating complex formations. The mathematical foundations of graph theory for multi-agent system control are introduced, and simulations are conducted to study the behaviour of the constructed model. Finally, some possible applications of this type of control are examined, and some open problems in the field are discussed, along some of the possible applicable upgrades to the model.

## Declaración Responsable sobre Autoría y Uso Ético de Herramientas de Inteligencia Artificial (IA)

Yo, YANRU WU JIN

Con DNI/NIE/PASAPORTE:

declaro de manera responsable que el/la presente:

- Trabajo de Fin de Grado (TFG)
- Trabajo de Fin de Máster (TFM)
- Tesis Doctoral

Titulado/a

### CONTROL DE SISTEMAS MULTIAGENTE Y SUS APLICACIONES

es el resultado de mi trabajo intelectual personal y creativo, y ha sido elaborado de acuerdo con los principios éticos y las normas de integridad vigentes en la comunidad académica y, más específicamente, en la Universidad Complutense de Madrid.

Soy, pues, autor del material aquí incluido y, cuando no ha sido así y he tomado el material de otra fuente, lo he citado o bien he declarado su procedencia de forma clara -incluidas, en su caso, herramientas de inteligencia artificial-. Las ideas y aportaciones principales incluidas en este trabajo, y que acreditan la adquisición de competencias, son mías y no proceden de otras fuentes o han sido reescritas usando material de otras fuentes.

Asimismo, aseguro que los datos y recursos utilizados son legítimos, verificables y han sido obtenidos de fuentes confiables y autorizadas. Además, he tomado medidas para garantizar la confidencialidad y privacidad de los datos utilizados, evitando cualquier tipo de sesgo o discriminación injusta en el tratamiento de la información.

En Madrid a 20/06/2024

# Índice

|  |           |
|--|-----------|
| <b>1. Introducción</b>   | <b>4</b>  |
| 1.1. Objetivos . . . . .                                       | 4         |
| <b>2. Definición del sistema</b>                               | <b>4</b>  |
| 2.1. Notación . . . . .  | 4         |
| 2.1.1. Formación nominal . . . . .                             | 5         |
| 2.1.2. Obtención de pesos $\omega_{ij}$ . . . . .              | 6         |
| <b>3. Dinámica de los agentes</b>                              | <b>7</b>  |
| 3.1. Entradas de control . . . . .                             | 7         |
| 3.2. Condición de estabilidad . . . . .                        | 8         |
| <b>4. Parámetros de movimiento</b>                             | <b>9</b>  |
| 4.1. Laplaciana modificada y transformaciones afines . . . . . | 9         |
| 4.2. Separación de componentes de la transformación . . . . .  | 10        |
| 4.3. Entrada de control con parámetros de movimiento . . . . . | 11        |
| <b>5. Resultados</b>   | <b>12</b> |
| 5.1. Caso bidimensional . . . . .                              | 12        |
| 5.1.1. Estabilización . . . . .                                | 12        |
| 5.1.2. Manejo afín . . . . .                                   | 14        |
| 5.2. Caso tridimensional . . . . .                             | 16        |
| 5.2.1. Estabilización . . . . .                                | 16        |
| 5.2.2. Manejo afín . . . . .                                   | 17        |
| <b>6. Retos, oportunidades y aplicaciones</b>                  | <b>18</b> |
| <b>7. Conclusión</b>   | <b>19</b> |

# 1. Introducción

Hoy en día el campo del control de sistemas se ha convertido en una de las disciplinas fundamentales y esenciales de la ingeniería y de la ciencia aplicada. Ejemplos como el sector aeronáutico, sistemas de transporte y de comunicaciones; todos requieren de un sistema de control para su correcto funcionamiento.

Dentro del control de sistemas, el control de enjambres es uno de los campos más fascinantes. Este se inspira, como indica su nombre, en los enjambres que se encuentran en la naturaleza [1], los cuales tienen una dinámica única capaz de controlar un grupo de agentes autónomos para que actúen como una única entidad coherente con unas instrucciones simples y locales seguidas por cada individuo. Algunos ejemplos podrían ser las bandadas de aves, bancos de peces o los enjambres de abejas.

## 1.1. Objetivos

Este trabajo se enfocará en construir un algoritmo capaz de guiar a un determinado número de agentes hacia una formación nominal y aplicar diferentes transformaciones afines a la formación, usando un enfoque *leaderless*, es decir, sin un líder de formación. Basándose en un sistema de control de tensiones, se construirá una matriz Laplaciana que contendrá los pesos adecuados asociados a las conexiones entre los diferentes agentes. La dinámica del sistema vendrá dada entonces por esta matriz, alcanzando el sistema una configuración de consenso. Además, se aplicarán transformaciones afines (ver Sección 4) a la formación, modificando la matriz Laplaciana con los parámetros de la transformación. Por último, se explorarán las potenciales aplicaciones de este tipo de sistemas en el mundo actual, desde uso militar hasta en la medicina [2].

# 2. Definición del sistema

## 2.1. Notación

Contamos con un número  $n \in \mathbb{N}$  de agentes móviles. Se denota  $\|x\|$  como la norma euclídea del vector  $x \in \mathbb{R}^p$ ,  $p \in \mathbb{N}$ , y dado un conjunto  $\mathcal{X}$ , su cardinalidad será definida como  $|\mathcal{X}|$ , esto es, el número de elementos del conjunto. El vector columna formado por todo unos se representará por  $\mathbf{1}_n$  y, dada una matriz  $A \in \mathbb{R}^{p \times q}$ ,  $p, q \in \mathbb{N}$ , se define el operador  $\bar{A} := A \otimes I_m \in \mathbb{R}^{pm \times qm}$ , donde  $\otimes$  es el producto de Kronecker y  $I_m \in \mathbb{R}^m$  es la matriz identidad.

Definiremos un grafo no dirigido como [3]  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ , donde  $\mathcal{V} = \{1, 2, \dots, n\}$ ,  $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V})$ , son conjuntos no vacíos de los nodos y pares de nodos sin ordenar (o aristas), respectivamente. Un grafo no dirigido es un grafo bidireccional donde las aristas no tienen una dirección de preferencia.

El conjunto  $\mathcal{N}_i$  contiene a los vecinos del nodo  $i$ , y está definido como  $\mathcal{N}_i := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ , lo cual indica que los vecinos de  $i$  son aquellos con los que está conectado por una arista de  $\mathcal{E}$ .

Sea  $\omega_{ij} \in \mathbb{R}$  el peso asociado al nodo  $(i, j) \in \mathcal{E}$ , de tal modo que la matriz Laplaciana<sup>1</sup>  $L \in \mathbb{R}^{n \times n}$  de  $G$  se define como:

$$l_{ij} := \begin{cases} \sum_{k \in \mathcal{N}_i} w_{ik} & \text{si } i = j, \\ -w_{ij} & \text{si } i \neq j \wedge j \in \mathcal{N}_i, \\ 0 & \text{si } i \neq j \wedge j \notin \mathcal{N}_i. \end{cases} \quad (1)$$

---

<sup>1</sup>Es frecuente en la literatura denominar a esta matriz, *stress matrix*, matrix de tensiones y reservar el nombre de matriz Laplaciana al caso en que  $\omega_{ij} = 1, \forall i, j$ . En este trabajo seguimos la definición de [3]

Asumimos que el grafo es conexo, esto es [4], para cada par de nodos  $(i, j) \in \mathcal{E}$ , existe al menos un camino (secuencia de aristas consecutivas) que los conecta, por lo que  $L\mathbf{1}_n = 0$ . Ya que tratamos con un grafo no dirigido, para cada par de nodos vecinos escogeremos una dirección arbitraria para construir un conjunto ordenado de aristas,  $\mathcal{Z}$ , de tal forma que para cada par de nodos vecinos  $\mathcal{Z}_k = (\mathcal{Z}_k^{\text{head}}, \mathcal{Z}_k^{\text{tail}})$ ,  $k \in \{1, \dots, \frac{|\mathcal{E}|}{2}\}$ , donde el primer elemento será la *cabeza* y el segundo, la *cola*. Esto será útil para construir una matriz de incidencia  $H \in \mathbb{R}^{|\mathcal{Z}| \times |\mathcal{V}|}$  que satisface también  $H\mathbf{1}_n = 0$ :

$$h_{ik} := \begin{cases} +1 & \text{si } i = \mathcal{Z}_k^{\text{tail}}, \\ -1 & \text{si } i = \mathcal{Z}_k^{\text{head}}, \\ 0 & \text{en otro caso.} \end{cases} \quad (2)$$

### 2.1.1. Formación nominal

Llamaremos *configuración* del conjunto de nodos  $n \in \mathcal{V}$  a sus coordenadas apiladas en el espacio Euclídeo  $p \in \mathbb{R}^{dn}$ ,  $p = [p_1^T, \dots, p_n^T]^T$ . Llamaremos estructura, *framework*, a un grafo equipado con una configuración, de tal forma que  $\mathcal{F} = (\mathcal{G}, p)$ . La figura 1 muestra un ejemplo de estructura.

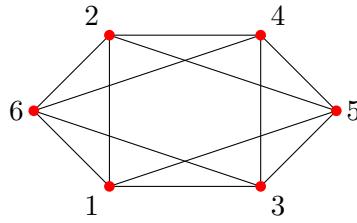


Figura 1: Ejemplo de *framework*

A partir de la configuración definiremos la *matriz de configuración*,  $P \in \mathbb{R}^{n \times d}$  y la *matriz de configuración aumentada*,  $\bar{P} \in \mathbb{R}^{n \times (d+1)}$  como:

$$P(p) = \begin{bmatrix} p_1^T \\ \vdots \\ p_n^T \end{bmatrix}, \quad \bar{P}(p) = \begin{bmatrix} p_1^T & 1 \\ \vdots & \vdots \\ p_n^T & 1 \end{bmatrix} = [P(p), \mathbf{1}_n] \quad (3)$$

Denotamos la formación nominal  $p^*$  para el grupo de agentes como:

$$p^* = (\mathbf{1}_n \otimes p_{\text{c.m.}}) + p_c^* \quad (4)$$

De tal manera que  $p_{\text{c.m.}}$  es la posición del centro de masas de la configuración y  $p_c^*$  son las coordenadas de los nodos desde el centro de masas, que dará a la formación su apariencia. En este trabajo se simplificará esto imponiendo  $p_{\text{c.m.}} = 0$ . Además asumiremos que  $p^*$  es genérica. Eso es [5], si todas las coordenadas  $p_1, \dots, p_n$  son algebraicamente independientes sobre los enteros, en otras palabras, no existe un polinomio no nulo con coeficientes enteros tal que  $f(p_1^1, \dots, p_1^m, \dots, p_n^1, \dots, p_n^m) = 0$ , donde  $p_i^j$  es el  $j$ -ésimo elemento del vector  $p_i$ .

Definimos [3] el concepto de forma deseada, construida desde la formación nominal de forma que la configuración estará en la forma deseada cuando:

$$p \in \mathcal{S} := \{p = (I_n \otimes A)p^* + (\mathbf{1}_n \otimes b) | A \in \mathbb{R}^{d \times d}, b \in \mathbb{R}^d\} \quad (5)$$

Por tanto,  $\mathcal{S}$  representa todas las posibles transformaciones afines de  $p^*$ .

### 2.1.2. Obtención de pesos $\omega_{ij}$

A continuación vamos a describir el método seguido para la obtención de los pesos de la Laplaciana. En un grafo no dirigido  $\mathcal{G}$  de  $n$  nodos y  $m$  aristas consideramos una dirección arbitraria para cada arista. Así, definimos la matriz de incidencia traspuesta  $H^T \in \mathbb{R}^{n \times m^2}$  [6] de forma que  $h_i \in \mathbb{R}^m$  sea la  $i$ -ésima columna de  $H^T$ , y por tanto  $H^T = [h_1, \dots, h_n]$ . Esto será lo que haga de la Laplaciana una matriz simétrica, ya que cada arista del grafo será asociado a un solo peso,  $\omega_{ij} = \omega_{ji}$ .

Definiendo una nueva matriz  $E \in \mathbb{R}^{n(d+1) \times m}$ :

$$E = \begin{bmatrix} \bar{P}^T(r)H\text{diag}(h_1) \\ \vdots \\ \bar{P}^T(r)H\text{diag}(h_n) \end{bmatrix} \quad (6)$$

Sea  $z_1 \dots z_q \in \mathbb{R}^m$  una base de  $\text{Null}(E)$ , el espacio nulo de la matriz  $E$ . El espacio nulo de una matriz  $A$  es el conjunto de soluciones tal que  $A\vec{x} = 0$ . Así, la base de este espacio estará compuesto por una combinación lineal de vectores que cumplan la condición. En este caso usaremos una descomposición en valores singulares (SVD) para obtener una base ortogonal de  $\text{Null}(E)$ . Supongamos que la SVD de una matriz  $A$  de rango  $k$  es  $A = U\Sigma V^T$ , una base ortogonal del espacio nulo de  $A$  vendrá dado por las últimas columnas de  $V$ , concretamente  $\vec{v}_{k+1}, \dots, \vec{v}_n$ .

A efectos prácticos se ha utilizado una función de la librería `numpy` para obtener la descomposición, y cogiendo la matriz  $V$  obtendremos la base buscada.

Ahora, para calcular los pesos  $\omega_{ij}$ , observemos la SVD de  $\bar{P} = U\Sigma V^T$ , ecuación (3), de modo que  $U = [U_1, U_2]$ , siendo  $U_1$  las primeras  $d + 1$  columnas de  $U$  y  $U_2$  el resto. De esta forma definiremos  $M_i = U_2^T H \text{diag}(z_i) H^T U_2$  para  $i = 1, \dots, q$ . Así, la matriz Laplaciana se puede calcular como:

$$L = H \text{diag}(\omega) H^T \quad (7)$$

Donde  $\omega$  se calcula como:

$$\omega = \sum_{i=1}^q c_i z_i, \quad (8)$$

donde  $c_1, \dots, c_q$  satisfacen la siguiente inecuación matricial lineal:

$$\sum_{i=1}^q c_i M_i \succ 0 \quad (9)$$

Para el cálculo de esta inecuación se ha usado `PICOS`, resultando en un *feasibility problem* (problema de factibilidad), el cual en nuestro caso es deducir si es factible encontrar unos valores de las constantes  $c_i$  que hacen de la suma en la ecuación 9 una matriz positiva semidefinida.

Vamos a calcular la matriz  $L$  para un cierto *framework*, en este caso el que aparece en la figura 1. Obtenemos:

$$L = \begin{bmatrix} 0.5 & 0.125 & -0.375 & 0 & 0.125 & -0.375 \\ 0.125 & 0.5 & 0 & -0.375 & 0.125 & -0.375 \\ -0.375 & 0 & 0.5 & 0.125 & -0.375 & 0.125 \\ 0 & -0.375 & 0.125 & 0.5 & -0.375 & 0.125 \\ 0.125 & 0.125 & -0.375 & -0.375 & 0.5 & 0 \\ -0.375 & -0.375 & 0.125 & 0.125 & 0 & 0.5 \end{bmatrix} \quad (10)$$

---

<sup>2</sup>Comparando la definición de matrices de incidencia de [3] y [6], resulta que están definidas como una siendo la traspuesta de la otra. Para simplificar la notación se ha escogido la definida en la ecuación 2 como la original y esta su traspuesta.

Visualizado en el grafo queda como se aprecia en la figura 2.

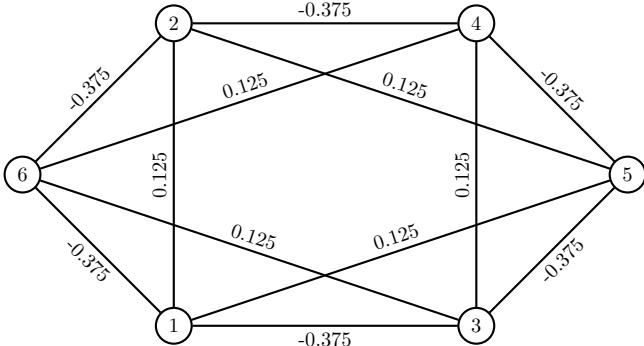


Figura 2: Framework y sus correspondientes pesos

### 3. Dinámica de los agentes

#### 3.1. Entradas de control

En nuestra formación, los agentes se mueven acorde a la matriz Laplaciana, la cual dará un valor de velocidad requerido a cada uno de ellos durante el tiempo de simulación. En este caso, consideraremos los agentes como masas puntuales, los cuales pueden moverse sin límites, es decir, no se simulará un caso real en el que normalmente se darían restricciones del propio agente (i.e. un rotor de dron no es capaz de generar un impulso mayor que el limitado por la potencia del motor). Una vez hecha la aclaración vamos a continuar definiendo la velocidad de los agentes [3].

$$\dot{p}_i = u_i, \quad i \in \mathcal{V} \quad (11)$$

Donde  $u_i \in \mathbb{R}^d$  será la entrada de control al agente  $i$ . De forma compacta:

$$\dot{p} = u \quad (12)$$

Donde  $p, u \in \mathbb{R}^{nd}$  son las posiciones y entradas de control apiladas. Dado que nos interesa que el control de la formación sea dado de forma local de forma que no sea necesaria una unidad central (control centralizado) sino que la toma de decisiones sea independiente en cada agente (control distribuido), la entrada la definiremos con las posiciones relativas de cada uno:

$$u_i = -h \sum_{j \in \mathcal{N}_i} \omega_{ij} (p_j - p_i) = -h \sum_{j \in \mathcal{N}_i} \omega_{zij} \quad (13)$$

Siendo  $h \in \mathbb{R}_+$ , una ganancia arbitraria positiva que podremos manipular para variar la velocidad con la que los agentes convergen a la formación nominal y, como hemos visto anteriormente,  $\omega_{ij} = \omega_{ji} \in \mathbb{R}$ . De forma compacta podemos escribir:

$$\dot{p} = -h \bar{L} p \quad (14)$$

Donde  $p(t) \rightarrow \text{Ker}\{\bar{L}\}$ . Las posiciones de la formación nominal cumplen  $Lp^* = 0$ , o lo que es lo mismo,  $p^* \in \text{Ker}\{\bar{L}\}$ , por lo que, para cada paso temporal las posiciones van a ir convergiendo hacia las posiciones buscadas. Entonces el kernel de  $\bar{L}$  es el conjunto  $\mathcal{S}$  resultante de forzar la condición:

$$\sum_{j \in \mathcal{N}_i} \omega_{ij} (p_i^* - p_j^*) = 0 \quad (15)$$

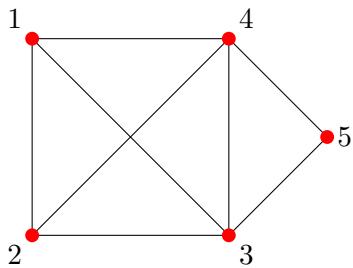
Donde los pesos calculados anteriormente construyen una matriz Laplaciana Buscaremos que esta sea semidefinida positiva (i.e. todos sus autovalores  $\geq 0$ ) y para ello se debe dar que el *framework* sea genéricamente y universalmente rígido, demostrado en el Teorema 2.2 de [5].

### 3.2. Condición de estabilidad

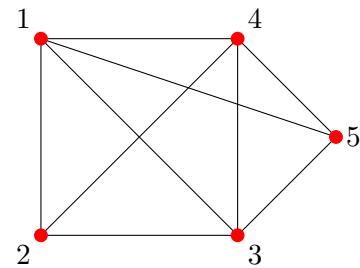
Dado un framework  $\mathcal{F} = (\mathcal{G}, p)$ , donde  $\mathcal{G}$  es un grafo no dirigido y  $p$  es genérica,  $p$  será estabilizable[5] sobre  $\mathcal{G}$  si y solo si  $\mathcal{G}$  es genéricamente y universalmente rígido. La condición genérica se cumple al asumir  $p$  genérica, mientras que la condición de rigidez universal se explica a continuación.

Dos *frameworks*  $(\mathcal{G}, p)$  en  $\mathbb{R}^{d_1}$  y  $(\mathcal{G}, q)$  en  $\mathbb{R}^{d_2}$  son equivalentes ( $\equiv$ ) si se conservan las distancias para los pares de nodos con aristas (se mantiene la escala). Dos *frameworks* serán congruentes ( $\cong$  si  $p$  y  $q$  se pueden obtener el uno del otro por un movimiento rígido como traslación o rotación (se mantiene la forma). Un *framework* en  $\mathbb{R}^d$  es universalmente rígido si para cualquier configuración  $q$  en  $\mathbb{R}^s$ ,  $s \in \mathbb{N}$ ,  $(\mathcal{G}, p) \equiv (\mathcal{G}, q)$  implica  $(\mathcal{G}, p) \cong (\mathcal{G}, q)$ .

Podemos realizar la prueba con el *framework* de la figura 3a, el cual no es universalmente rígido ya que en  $\mathbb{R}^3$ , el triángulo  $\Delta 345$  se puede “plegar” sobre la arista  $(3,4)$ , incumpliendo la condición dada de rigidez universal. Con la ayuda de las simulaciones, cuyos detalles veremos en secciones posteriores, (figura 4a) nos damos cuenta de que el agente 5 no se estabiliza con el resto. Añadiendo una arista más (fig 3b),  $(1,5)$  o  $(2,5)$ , se cumpliría con la condición de rigidez, por lo que el sistema sí sería estabilizable (figura 4b).

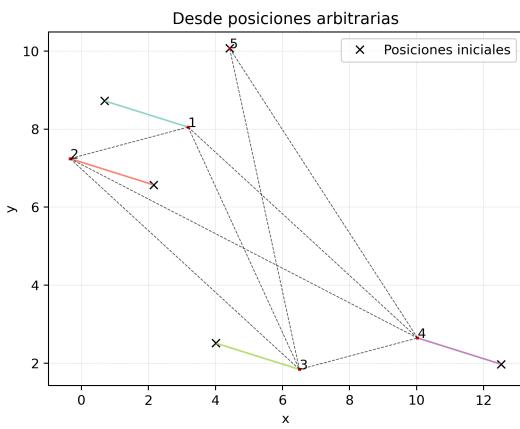


(a) *Framework* no universalmente rígido

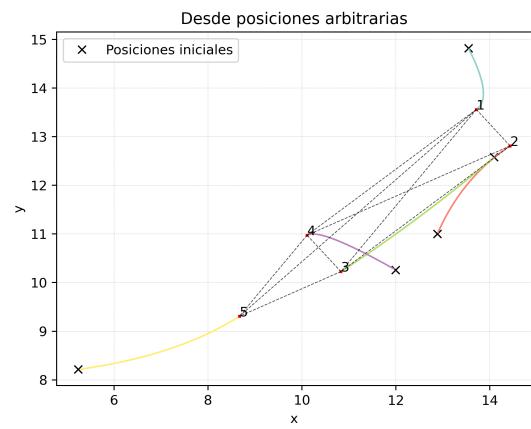


(b) *Framework* universalmente rígido

Figura 3: *Frameworks* y manipulación de rigidez



(a) Simulación con *framework* no rígido.



(b) Simulación con *framework* rígido

Figura 4: Simulaciones según rigidez

## 4. Parámetros de movimiento

### 4.1. Laplaciana modificada y transformaciones afines

Una vez hemos conseguido definir una forma de estabilizar el sistema hacia una formación vamos a modificar la Laplaciana para que el sistema sea capaz de ejecutar transformaciones afines, que son transformaciones que conservan la colinealidad (puntos colineales se mantienen colineales) y paralelismos, aunque no necesariamente ángulos ni distancias. En este trabajo se estudiarán 4: traslación, rotación, escalado y cizallamiento, como se muestran en la figura 5. Nuestro objetivo será modificar los pesos  $\omega_{ij}$ , y por tanto la matriz Laplaciana  $L$ , de tal forma que para  $t \rightarrow \infty$ ,  $p(t)$  converja a unas trayectorias de movimiento estable.

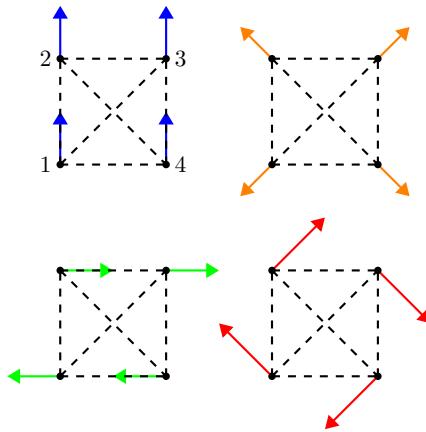


Figura 5: Transformaciones afines (traslación, escalado, cizallamiento y rotación)

Consideremos los pesos modificados [3],  $\tilde{\omega}_{ij}$ , tal que:

$$\tilde{\omega}_{ij} = \omega_{ij} - \frac{\kappa}{h} \mu_{ij} \quad (16)$$

Donde  $\mu_{ij} \in \mathbb{R}$  serán los parámetros del movimiento que corresponderán a las diferentes transformaciones,  $\kappa \in \mathbb{R}$  será una ganancia arbitraria que dictará la velocidad del movimiento de la transformación dada y el término  $h$  es la misma ganancia que en la ecuación 14, que se introduce para compensar que al nuevo Laplaciano modificado se le volverá a multiplicar esta cantidad. En el enfoque de control distribuido de este trabajo, esto indica que  $\mu_{ij} = 0$  si  $j \notin \mathcal{N}_i$ , y  $\mu_{ij} \neq \mu_{ji}$ .

Al igual que con la matriz de incidencia, asignaremos con la dirección arbitraria un valor positivo y otro negativo para una nueva matriz  $M \in \mathbb{R}^{n \times m}$ , cuyos elementos serán:

$$m_{ik} := \begin{cases} +\mu_{i\mathcal{Z}_k^{\text{head}}} & \text{si } i = \mathcal{Z}_k^{\text{tail}}, \\ -\mu_{i\mathcal{Z}_k^{\text{tail}}} & \text{si } i = \mathcal{Z}_k^{\text{head}}, \\ 0 & \text{en otro caso.} \end{cases} \quad (17)$$

Así podremos construir un Laplaciano modificado con los nuevos parámetros de movimiento a través de  $M$ ,

$$\tilde{L} = L - \frac{\kappa}{h} M H \quad (18)$$

Y usando este nuevo Laplaciano con la ecuación 14, obtendremos nuestras entradas de control de cada agente. Para ello vamos a ver cómo obtener los parámetros [3]. De forma similar a los pesos  $\omega_{ij}$ ,

$${}^b v_i^* = \sum_{j \in \mathcal{N}_i} \mu_{ij} \left( {}^b p_i^* - {}^b p_j^* \right) = \sum_{j \in \mathcal{N}_i} \mu_{ij} {}^b z_{ij}^*, \quad \forall i \in \mathcal{V}, \quad (19)$$

donde  ${}^b v_i^* \in \mathbb{R}^d$  es la velocidad para cada agente  $i$  tal que el movimiento colectivo sea compatible con  $p \in \mathcal{S}$  (ecuación 5). El superíndice  $b$  es simplemente un indicativo de la toma de origen de coordenadas  $O_b$  sobre el centro de masa de la formación nominal  $p^*$ . En la definición se usan, una vez más, las posiciones relativas entre agentes  $z^* = \bar{H}^T p^*$ , por lo que si todos los vectores apilados de  $z^*$  sufren una transformación afín, el movimiento resultante se transformará de igual forma. Así pues, por ejemplo, partiendo de un movimiento circular alrededor del centro de masas de  $p^*$ , una vez apliquemos una transformación afín (e.g. una traslación), el movimiento se transformará de igual forma, incorporando la transformación a este y generando trayectorias elipsoidales en vez de las circulares originales. La figura 6 muestra el ejemplo.

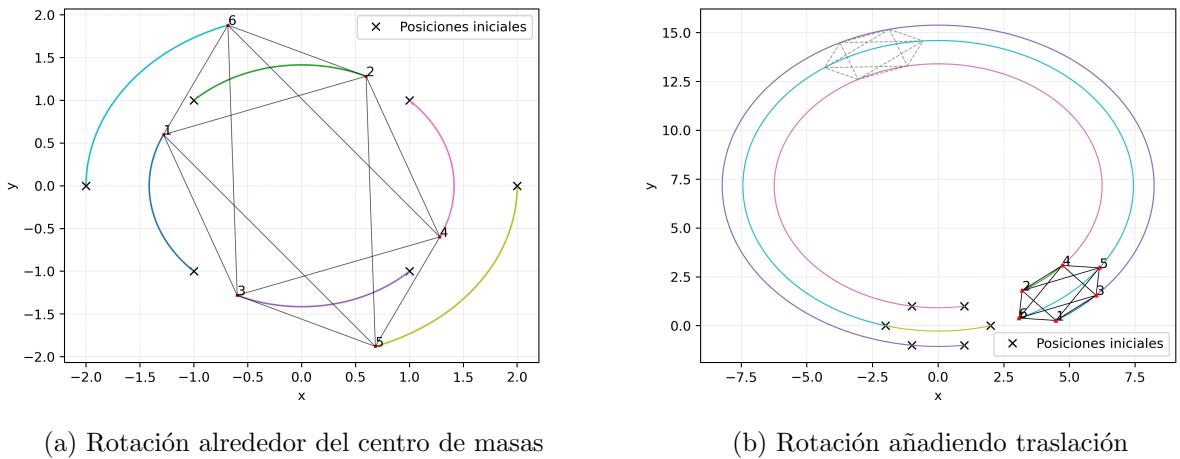


Figura 6: Ejemplo de transformación de trayectorias

Vamos a analizar un par de casos de ejemplo para entender mejor cómo calcular estos parámetros. Digamos que queremos aplicar una traslación vertical hacia arriba, como la que se muestra en la figura 5. Entonces queda claro que las velocidades de todos los agentes deben ser  ${}^b v_i^* = [0, 1]^T$ , para  $i = 1, 2, 3, 4$ . Considerando que los lados de la formación miden 1, para el agente 1 las posiciones relativas serán  $(p_1^* - p_2^*) = [0, -1]^T$ ,  $(p_1^* - p_3^*) = [-1, -1]^T$ ,  $(p_1^* - p_4^*) = [-1, 0]^T$ , por lo que para satisfacer la ecuación 19 podemos escoger  $\mu_{13} = \mu_{12} = 0$ ,  $\mu_{14} = -1$ , por lo que

$${}^b v_1^* = \mu_{12}(p_1^* - p_2^*) + \mu_{13}(p_1^* - p_3^*) + \mu_{14}(p_1^* - p_4^*) = -(p_1^* - p_4^*) = -[1, 0]$$

Podemos formular en forma compacta:

$${}^b v_f^* = \overline{MH} {}^b p^* \quad (20)$$

Con  ${}^b v_f^*$  las velocidades de los agentes apiladas.

## 4.2. Separación de componentes de la transformación

Para simplificar la manipulación del sistema podemos separar las velocidades en componentes de las diferentes transformaciones [3], por lo que en  $\mathbb{R}^2$  habrá 6 componentes del movimiento diferentes, dividiendo  $M$  en:

$$M = \kappa_{t_1} M_{t_1} + \kappa_{t_2} M_{t_2} + \kappa_s M_s + \kappa_r M_r + \kappa_{s_1} M_{s_1} + \kappa_{s_2} M_{s_2} \quad (21)$$

Donde  $t_1, t_2$  representan las traslaciones en cada eje,  $s$  el escalado,  $r$  las rotaciones y  $s_1, s_2$  el cizallamiento sobre cada eje y todas  $\in \mathbb{R}^{m \times n}$ . Las constantes  $\kappa$  de cada transformación serán las ganancias, indicando las “coordenadas” del movimiento, que en la práctica indicaran la velocidad a la que se desarrolla cada tipo de trasnformación afín.

Vamos a definir cada componente de  $M$  a partir de la ecuación 20 (demonstrado en *Lemma 1* de [3]). Primero, la traslación será directamente:

$$\sum_{l=1}^d \bar{M}_{t_l} \bar{H} p^* = (\mathbf{1}_n \otimes v^*), \quad (22)$$

Donde  $v^* \in \mathbb{R}^d$  será la velocidad común de la traslación. Es sencillo darse cuenta que esta componente es la misma que aquella expresada previamente en la ecuación 5, donde se veía que la forma deseada podría ser simplemente una traslación con  $v^* = b$ .

En cuanto a las rotaciones, estas vendrán dadas por:

$$\bar{M}_r \bar{H} p^* = (I_n \otimes W) p^* \quad (23)$$

Donde  $W \in \mathbb{R}^{d \times d}$  es el tensor de velocidad angular:

$$\begin{array}{ll} \text{2D : } & \left[ \begin{array}{cc} 0 & -\omega \\ \omega & 0 \end{array} \right] \\ & \text{3D : } \left[ \begin{array}{ccc} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{array} \right] \end{array}$$

Donde ahora  $\omega$  representa las velocidades angulares sobre cada uno de los ejes cartesianos. En el caso 2D solo hay un posible eje de rotación mientras que en 3D existen 3.

Para el escalado, esta seguirá simplemente:

$$\bar{M}_s \bar{H} p^* = p^* \quad (24)$$

Ya que las posiciones de los agentes desde la formación nominal solo sufrirán un escalado constante sobre sus coordenadas, el cual vendrá dado al definir la ganancia  $\kappa_s$ .

Y por último, el cizallamiento vendrá dado por:

$$\sum_{l=1}^d \bar{M}_{s_l} \bar{H} p^* = (I_n \otimes S), \quad (25)$$

Donde, de forma similar a las rotaciones,  $S \in \mathbb{R}^{d \times d}$  será el tensor de cizallamiento, que para 2D y 3D:

$$\begin{array}{ll} \text{2D : } & \left[ \begin{array}{cc} 0 & h_{xy} \\ h_{yx} & 0 \end{array} \right] \\ & \text{3D : } \left[ \begin{array}{ccc} 0 & h_{xy} & h_{xz} \\ h_{yx} & 0 & h_{yz} \\ h_{zx} & h_{zy} & 0 \end{array} \right] \end{array}$$

Donde  $h$  ahora indica la velocidad de cizallamiento paralelo a los ejes indicados.

### 4.3. Entrada de control con parámetros de movimiento

Una vez hemos definido la matriz  $M$  en componentes, la dinámica del sistema modificado se puede interpretar fácilmente, de tal forma que introduciendo los parámetros que deseemos el sistema será simulable con un control iterativo sobre el tiempo donde las posiciones varían de forma que:

$$p(t) = p(t - dt) + \dot{p}(t)dt \quad (26)$$

Pero para ello debemos introducir los nuevos pesos modificados en la expresión de  $\dot{p}$ . Usando 14 y 18:

$$\dot{p} = -h\tilde{\bar{L}}p = -h\bar{L}p - \kappa\bar{M}\bar{H}p \quad (27)$$

Se puede estudiar la estabilidad de Lyapunov del sistema descrito, Teorema 1 de [3], para obtener que para  $t \rightarrow \infty$ ,  $p(t) \rightarrow \mathcal{S}$  y  $\dot{p}(t) \rightarrow \kappa\bar{M}\bar{H}p(t)$  simultáneamente.

## 5. Resultados

Las simulaciones que se muestran a continuación demuestran el buen funcionamiento del modelo en 2 y 3 dimensiones, tanto la estabilización de los agentes como la aplicación de parámetros de movimiento de transformaciones afines. Se incluye además un caso con un número de agentes significativamente mayor, para mostrar la capacidad de implementar este mismo sistema de control a enjambres más numerosos.

### 5.1. Caso bidimensional

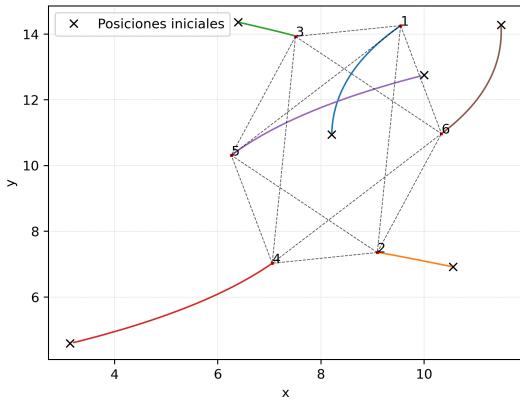
#### 5.1.1. Estabilización

Para el caso 2D, se ha tomado el *framework* descrito en la figura 1. El grafo se describe con 6 nodos y 12 aristas de modo que:

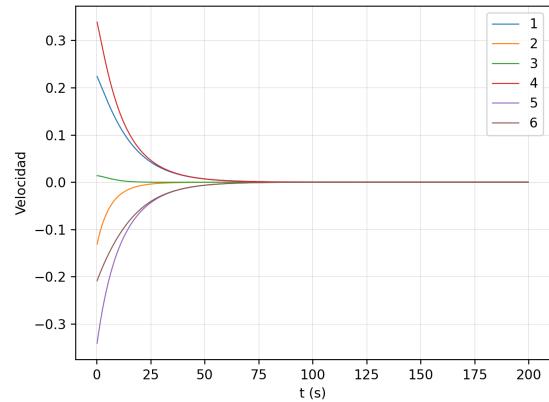
$$\mathcal{V} = \{1, 2, 3, 4, 5, 6\}$$

$$\mathcal{Z} = \{(1, 2), (1, 3), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6)\}$$

Los pesos asociados son los que componen  $L$  en 10 y los mostrados en la figura 2. El grafo escogido cumple la condición de estabilidad tal que es genéricamente y universalmente rígido y, adicionalmente, se podría eliminar una de las aristas (e.g. (2, 5)) y se mantendría la estabilidad en el sistema. Al igual que se observa que  $p(t) \rightarrow \mathcal{S}$  (figura 9a) mientras  $\dot{p} \rightarrow 0^3$  (figura 9b) para  $t \rightarrow \infty$ , la figura 9 muestra el mismo comportamiento para el nuevo *framework*.



(a) Estabilización desde posiciones arbitrarias



(b) Velocidad de los agentes

Figura 7: Estabilización de los agentes

---

<sup>3</sup>Para la obtención numérica de las velocidades se ha empleado una aproximación de la derivada  $\frac{dr}{dt} \approx \frac{\Delta r}{\Delta t} = \dot{r}$

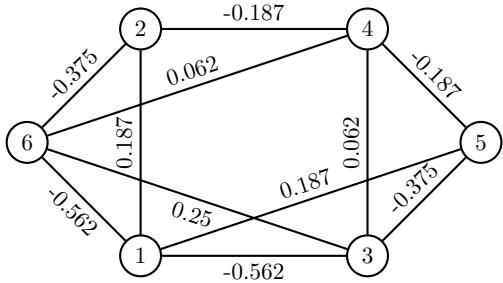


Figura 8: Framework y sus correspondientes pesos

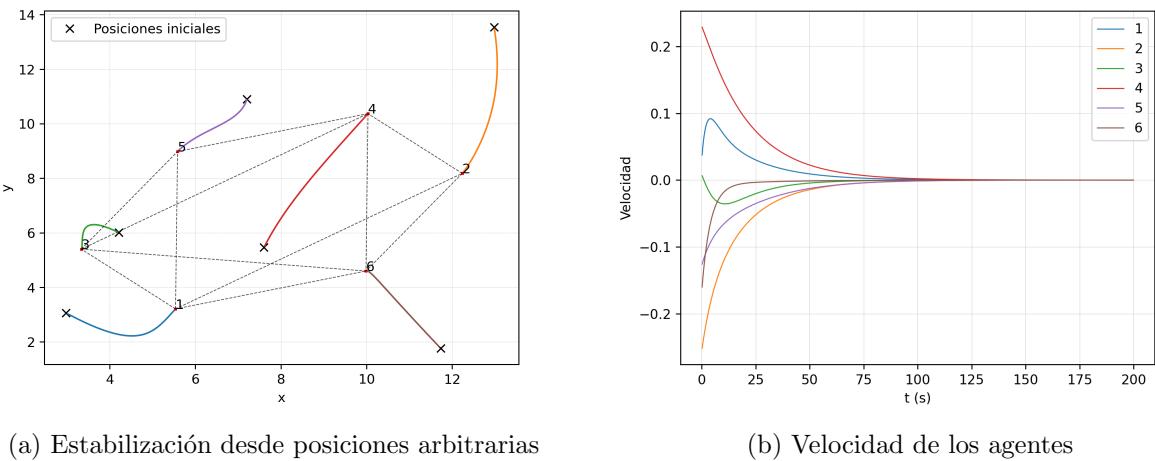


Figura 9: Estabilización de los agentes sobre el *framework* original con una arista menos

También podemos modificar la ganancia  $h$  descrita en la ecuación 14. En las simulaciones anteriores se ha empleado  $h = 0.1$ . La magnitud de esta constante indicará la velocidad con la que los agentes convergerán  $p \rightarrow S$ . Se observa cómo aumentándolo los agentes tardan menos en llegar a  $S$  en la figura 10.

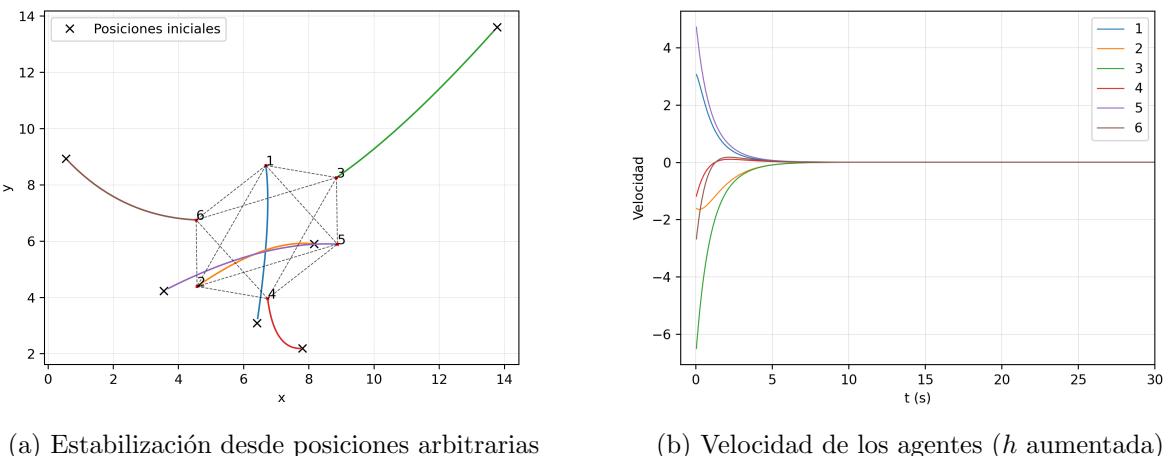


Figura 10: Estabilización de los agentes con ganancia  $h = 1$

Analizando 10b, los agentes convergen hacia  $S$  en un tiempo mucho menor que en 9b (un orden de magnitud menor). Como consecuencia las velocidades de éstos también es mucho mayor, lo cual en un caso real, dependiendo de las características de los agentes, podría no ser posible de ejecutar.

También se han realizado simulaciones con un número significativamente mayor de agentes, como el caso de la configuración de la figura 11, un polígono de  $n = 30$ . Concretamente se ha generado un polígono estrellado unido con  $k = 15$ , siendo  $k$  el *step* de unión de nodos. Esto lleva a una estructura donde cada nodo está unido solo a 3 vecinos.

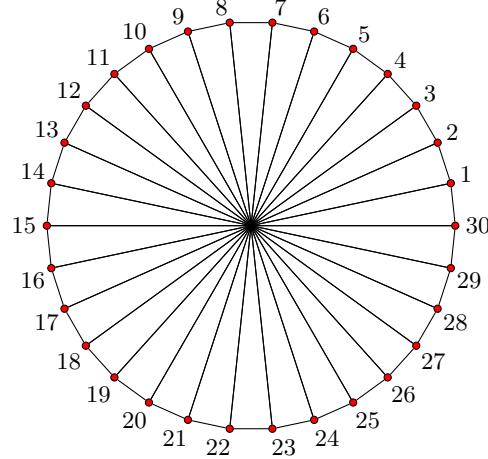


Figura 11:  $n = 30$ ,  $k = 15$

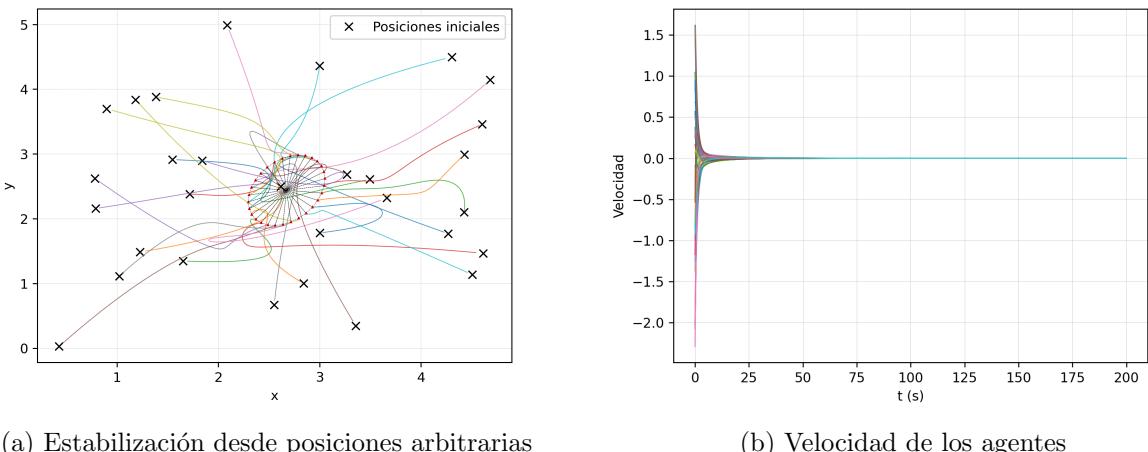
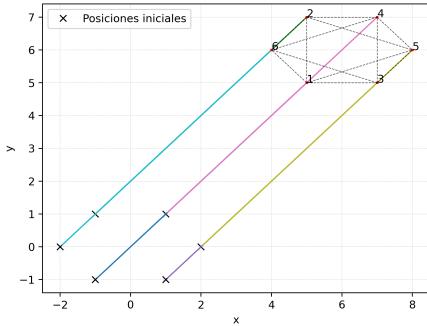


Figura 12: Estabilización de los agentes con ganancia  $h = 0.8$

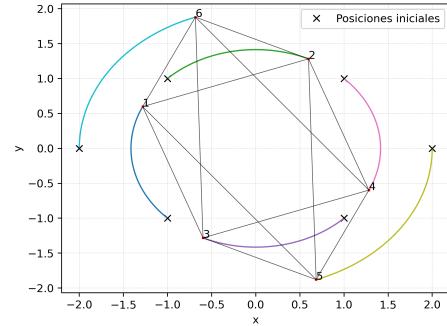
Se observa que la estabilización se lleva a cabo exitosamente, siendo una muestra de control distribuido muy eficiente.

### 5.1.2. Manejo afín

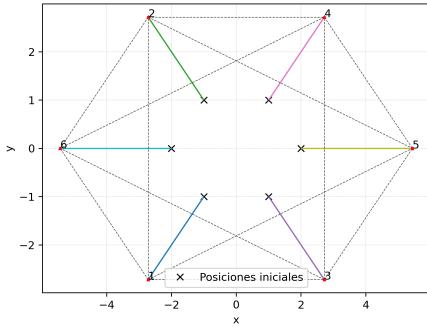
En cuanto a los parámetros de control, podemos simular cada una de las transformaciones mencionadas de forma independiente, mostradas en la figura 13.



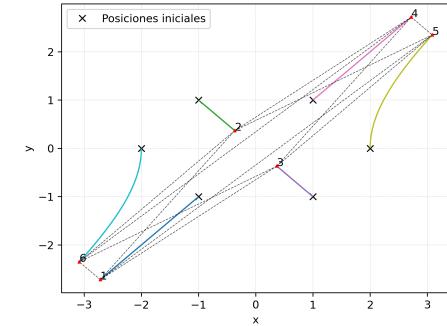
(a) Traslación con  $v = [1 \ 1]$



(b) Rotación:  $\omega = 1$



(c) Escalado positivo



(d) Cizallamiento sobre ambos ejes:  $h_{xy} = h_{yx} = 1$

Figura 13: Transformaciones afines

Podemos ahora construir una simulación ejemplo de un uso de los parámetros de movimiento junto a la estabilización provista por el control por pesos del Laplaciano. Modificando  $L \rightarrow \tilde{L}$  como descrito en 18 con unos parámetros prefijados y aplicando 27 obtenemos la figura 14. Las ganancias son  $h = 1$ ,  $\kappa_t = 2$ ,  $\kappa_s = -0.006$ ,  $\kappa_r = 1$ , sin cizallamiento.

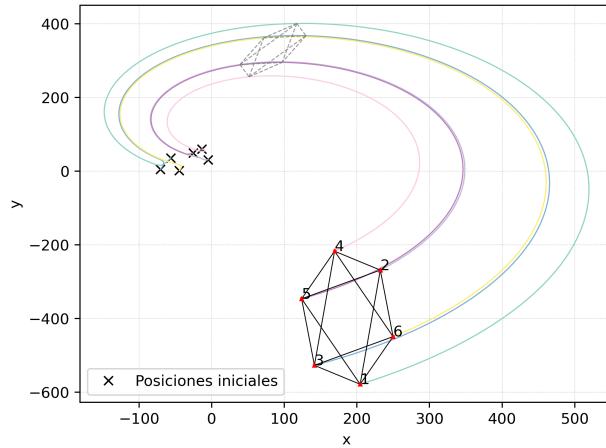


Figura 14:  $v = [1 \ 0]$ ,  $\omega = 1$

## 5.2. Caso tridimensional

Construyamos un nuevo *framework*, el cual cumplirá las condiciones de estabilidad (genérica y universalmente rígido). Este vendrá definido por el grafo:

$$\mathcal{V} = \{1, 2, 3, 4, 5, 6\}$$

$$\mathcal{E} = \{(1,2), (1,3), (1,4), (1,5), (1,6), (2,3), (2,4), (2,5), (2,6), (3,4), (3,5), (3,6), (4,5), (4,6), (5,6)\}$$

Y la configuración será tal que los nodos estén posicionados alrededor del centro en  $(0,0,0)$ , tal como muestra la figura 15.

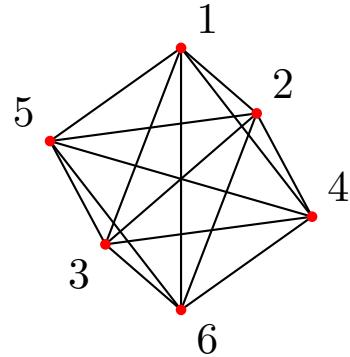


Figura 15: *Framework* 3D

### 5.2.1. Estabilización

En este caso el *framework* empleado está totalmente conectado, lo cual nos asegura que cumple las condiciones de estabilidad. Aunque en un caso práctico no sería lo ideal, nos servirá como ejemplo y para demostrar el funcionamiento del modelo de control sobre configuraciones tridimensionales.

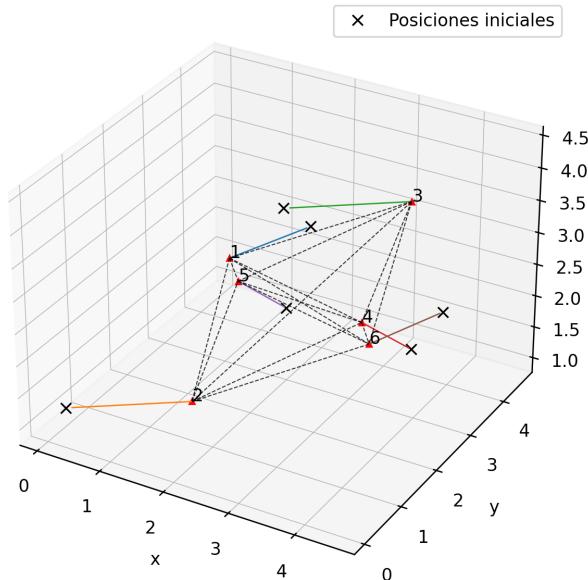


Figura 16: Estabilización desde posiciones arbitrarias

### 5.2.2. Manejo afín

Hemos de tener en cuenta que en este caso contamos con un grado de libertad adicional, lo cual amplía las posibilidades de movimiento en casi todos los casos (exceptuando el escalado que puede ser positivo o negativo): la translación tiene una dirección más, la rotación 2 ejes más y el cizallamiento añade un eje, aumentando el número de elementos de la matriz  $S$  (25) a 6.

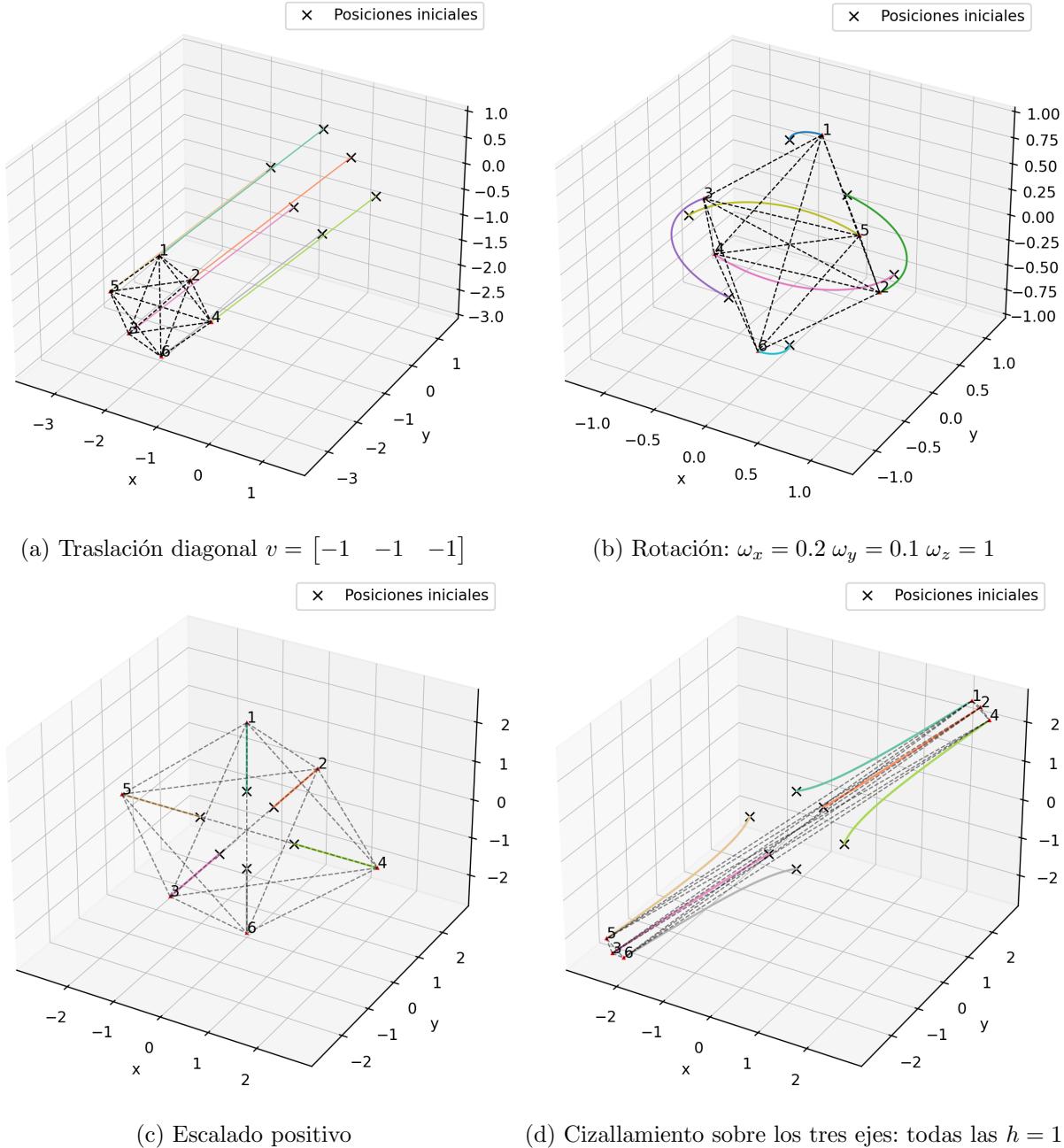


Figura 17: Transformaciones afines en 3D

Realizamos una simulación a partir de unas posiciones arbitrarias de los agentes con unos parámetros y ganancias prefijados de distintos tipos de transformaciones, la cual se muestra en la figura 18. Las ganancias en este caso son  $h=0.5$ ,  $\kappa_t=-0.25$ ,  $\kappa_s=-0.005$ ,  $\kappa_r=1$ , sin cizallamiento.

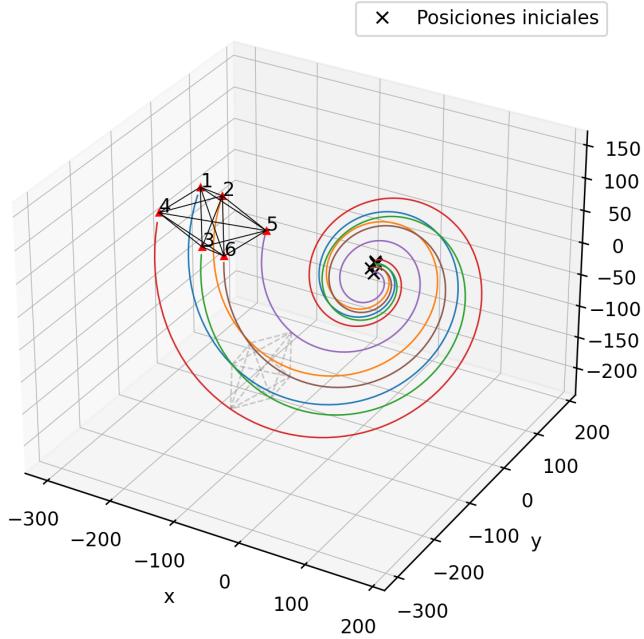


Figura 18:  $\omega_x = 0, \omega_y = -1, \omega_z = 1, v = [1 \ 1 \ 1]$

## 6. Retos, oportunidades y aplicaciones

En esta última sección vamos a abordar algunas de las limitaciones y problemas abiertos que existen con este tipo de control de sistemas, además de algunas ideas de implementación y mejora del modelo.

A lo largo de este trabajo se han observado algunos potenciales problemas que podrían surgir a la hora de aplicar este sistema de control a un entorno real, como el problema de las limitaciones de los agentes en cuanto a la saturación de velocidades, donde el agente podría no ser capaz de ejecutar las acciones impuestas por el modelo. Además, las trayectorias están limitadas a aquellas dictadas por las posiciones iniciales y los parámetros de control. Esto significa que no se pueden dirigir manualmente por trayectorias predefinidas, como en el modelo de control descrito en [7], por lo que sería interesante tratar de combinar estos dos enfoques.

Otra posible limitación es la robustez frente a fallos de los agentes. Si bien los sistemas distribuidos ofrecen cierta resiliencia, la pérdida de un agente en la red puede afectar significativamente la formación y el control del grupo, por lo que tratar de implementar un sistema de reconfiguración dinámica que reorganice a los agentes restantes en una formación sería ideal, evitando que la pérdida de agentes comprometa la integridad de la misión.

En cuanto a las aplicaciones, este tipo de sistema de control descentralizado podría resultar ser muy prácticos en situaciones que estén limitadas por potencia de procesamiento, como algunos casos que describe [8], este podría ser empleado como método de exploración en entornos complicados. Se exploran los potenciales usos en UAV (vehículo aéreo no tripulado), UGV (vehículo terrestre no tripulado), USV (vehículo de superficie) y AUV (vehículo submarino autónomo) para la exploración. Concretamente, el uso en entornos subacuáticos muestra gran potencial, al tratarse de áreas casi inaccesibles con vehículos tripulados, además de tener comunicaciones limitadas debido a las condiciones subacuáticas, por lo que un control más distribuido sería de gran ayuda. En la práctica, hoy en día vemos este tipo de control en proyectos [9] como SWARMIES de la NASA

para exploración de superficies extraterrestres, y en otros como la plataforma de investigación *Distributed Flight Array* de coordinación de rotores individuales en entornos aéreos.

Otras aplicaciones más inesperadas son aquellas en el área de medicina, como ya fue mencionado al inicio, como por ejemplo [2] menciona un método de identificación de metástasis a partir de escaneos óseos, usando un sistema de inteligencia de enjambre.

## 7. Conclusión

En este trabajo, hemos abordado el problema del control de formaciones afines en sistemas multiagente, desarrollando e implementando algoritmos de control distribuido que permiten a los agentes alcanzar y mantener configuraciones deseadas. Se ha demostrado la eficacia de estos algoritmos en la gestión y coordinación de formaciones complejas, sin requerir de una unidad central de control, haciendo de este sistema uno mucho más eficiente y de menor requerimiento de procesamiento que en el caso de un control centralizado.

Hemos revisado las bases matemáticas de la teoría de grafos, ahondando en la aplicación de esta a redes multiagente: realimentación basada en grafos pesados, matrices de estrés, matrices laplacianas, entre otros temas.

Todo ello ha sido realizado a través de código en `python`, haciendo uso de sus numerosas librerías de cálculo matemático, matricial e incluso específicas para problemas de desigualdades matriciales. De igual modo, las simulaciones llevadas a cabo en este trabajo fueron generadas por ordenador en un entorno de `python`, que ha demostrado ser más que suficiente para aquello que hemos querido abordar, siendo una útil herramienta para demostrar la eficacia de este método de control de enjambres.

Por último se han visto algunas de las posibles mejoras del modelo, de manera que el control sea más robusto y adaptable a las necesidades y problemas que puedan presentarse en un caso real.

## Referencias

- [1] Y. Tan y Z.-y. Zheng, “Research Advance in Swarm Robotics”, *Defence Technology*, vol. 9, n.<sup>o</sup> 1, págs. 18-39, 2013, ISSN: 2214-9147. DOI: <https://doi.org/10.1016/j.dt.2013.03.001>. dirección: <https://www.sciencedirect.com/science/article/pii/S221491471300024X>.
- [2] M. M. al-Rifaie, A. Aber y R. Raisys, “Swarming Robots and Possible Medical Applications”, en *International Society for the Electronic Arts (ISEA 2011)*, Istanbul, Turkey, 2011. dirección: <https://research.gold.ac.uk/id/eprint/17283/>.
- [3] H. G. de Marina, J. Jimenez Castellanos y W. Yao, “Leaderless collective motions in affine formation control”, en *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, págs. 6433-6438. DOI: [10.1109/CDC45484.2021.9682800](https://doi.org/10.1109/CDC45484.2021.9682800).
- [4] M. Mesbahi y M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, STU - Student edition. Princeton University Press, 2010, págs. 15-16, ISBN: 9780691140612.
- [5] Z. Lin, L. Wang, Z. Chen, M. Fu y Z. Han, “Necessary and Sufficient Graphical Conditions for Affine Formation Control”, *IEEE Transactions on Automatic Control*, vol. 61, n.<sup>o</sup> 10, págs. 2877-2891, 2016. DOI: [10.1109/TAC.2015.2504265](https://doi.org/10.1109/TAC.2015.2504265).
- [6] S. Zhao, “Affine Formation Maneuver Control of Multiagent Systems”, *IEEE Transactions on Automatic Control*, vol. 63, n.<sup>o</sup> 12, págs. 4140-4155, 2018. DOI: [10.1109/TAC.2018.2798805](https://doi.org/10.1109/TAC.2018.2798805).
- [7] W. Yao, H. G. de Marina, Z. Sun y M. Cao, “Distributed coordinated path following using guiding vector fields”, en *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, págs. 10 030-10 037.
- [8] L. Briñón-Arranz, A. Renzaglia y L. Schenato, “Multi-Robot Symmetric Formations for Gradient and Hessian Estimation with Application to Source Seeking”, *IEEE Transactions on Robotics*, vol. PP, feb. de 2019. DOI: [10.1109/TR0.2019.2895509](https://doi.org/10.1109/TR0.2019.2895509).
- [9] M. Schranz, M. Umlauft, M. Sende y W. Elmenreich, “Swarm Robotic Behaviors and Current Applications”, *Frontiers in Robotics and AI*, vol. 7, 2020, ISSN: 2296-9144. DOI: [10.3389/frobt.2020.00036](https://doi.org/10.3389/frobt.2020.00036). dirección: <https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2020.00036>.